IEEE TITLE*COMP ** 4805702 0052741 5

IEEE Standard

Computer Dictionary

Compilation of IEEE Standard Computer Glossaries

610

computer-aided engineering (CAE). The use of computers to aid in e
neering analysis and design. May include solution of mathematical
lems, process control, numerical control, and execution of prograr
forming complex or repetitive calculations. *See also:* **compute**
design; computer-aided manufacturing. [610.2]

computer language. A language designed to enable humans
cate with computers. *See also:* **design language; query la**
gramming language. [610.12]

computer program. A combination of computer instruct
nitions that enable computer hardware to perform com
functions. *See also:* **software.** [610.5, 610.12]

computer software configuration item (CSCI).
ware that is designated for configuration manage
gle entity in the configuration management pr
ware configuration item. *See also:* **configur**

computer system. A system containing one
ated software.

computer word. *See:* **word.**

computing center. A facility desig
ty of users through the opera
h services provided b

Published by the Institute of
Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394 USA.

IEEE

SH13847 January 18, 1991

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052742 7

IEEE Standard Computer Dictionary

A Compilation of IEEE Standard Computer Glossaries

610

Sponsor

Standards Coordinating Committee
of the
IEEE Computer Society

Abstract: *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries,* identifies terms currently in use in the computer field. Standard definitions for those terms are established.
Keywords: Glossary; terminology; definitions; dictionary.

Library of Congress Catalog Number 90-086306

ISBN 1-55937-079-3

Copyright © 1990 by

The Institute of Electrical and Electronics Engineers
345 East 47th Street, New York, NY 10017, USA

No part of this document may be reproduced in any form,
in an electronic retrieval system or otherwise,
without the prior written permission of the publisher.

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE which have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least once every five years for revision or reaffirmation. When a document is more than five years old, and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason IEEE and the members of its technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
345 East 47th Street
New York, NY 10017
USA

IEEE Standards documents are adopted by the Institute of Electrical and Electronics Engineers without regard to whether their adoption may involve patents on articles, materials, or processes. Such adoption does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the standards documents.

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

Foreword

(This Foreword is not a part of the IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries.)

The computer field is continuing to expand. New terms are being generated and new meanings are being adopted for existing terms. The IEEE Computer Dictionary project was undertaken to document this vocabulary. Its purpose is to identify terms currently in use in the computer field and to establish standard definitions for these terms. The dictionary is intended to serve as a useful reference for those in the computer field and for those who come into contact with computers either through their work or in their everyday lives.

The dictionary is being developed as a series of subject-area glossaries, each approved and published as it is completed. This edition of the dictionary contains all terms and definitions from the glossaries published to date. The IEEE designations and subject matter of these glossaries are as follows:

IEEE Std 1084 [610.1] -1986	Mathematics of Computing Terminology
IEEE Std 610.2-1987	Computer Applications Terminology
IEEE Std 610.3-1989	Modeling and Simulation Terminology
IEEE Std 610.4-1990	Image Processing and Pattern Recognition Terminology
IEEE Std 610.5-1990	Data Management Terminology
IEEE Std 610.12 [729] -1990	Software Engineering Terminology

The remaining glossaries in the series are listed below. Future editions of the dictionary will contain terms and definitions from those glossaries that have been published at the time of dictionary compilation.

P610.6	Computer Graphics Terminology
P610.7	Computer Networking and Data Communications Terminology
P610.8	Artificial Intelligence Terminology
P610.9	Computer Security and Privacy Terminology
P610.10	Computer Hardware Terminology
P610.11	Theory of Computation Terminology
P610.13	Computer Languages Terminology

Every effort has been made to use definitions from established standards in this dictionary. When existing standards were found to be incomplete, unclear, or inconsistent with other entries in the dictionary, however, new, revised, or composite definitions have been developed.

At the time of publication of this document, the following people form the steering committee of the Computer Dictionary working group:

Anne Geraci	Louise McMonegal	Dr. Fredrick Springsteel
Freny Katki	Bennett Meyer	Paul Wilson
Dr. John Lane	Dr. Hugh Porteous	Mary Yee
	Jane Radatz	

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

Other working group members who contributed to the subject glossaries were as follows:

610.1 [1084-1986]

H. Ronald Berlack
William M. Dupras
Joel J. Forman
John A. Goetz
Virl Haas
Paul Heinzelman
Leonard W. Seagren
Michael J. Strait

610.2-1990

Russell J. Abbott
A. Frank Ackerman
H. Ronald Berlack
Kathleen L. Briggs
Homer C. Carney
Katherine Coleman
Joel J. Forman
Gary R. Gladden
John B. Goetz
Virl Haas
Richard Landry
Gary S. Lindsay
Rollin P. Mayer
William E. Perry
Lawrence W. Schear
Leonard W. Seagren
Wayne Smith
Elias Tonias
Andrew H. Weigel
W. Martin Wong

610.3-1990

Jerry Banks
H. Ronald Berlack
James Bradley
William Dupras
John Goetz
Virl Haas
Walter Karplus
Michael Kong
Gary Lindsay
Philip Marriott
Russell Mikel
Thomas Perrone
Donald Phillips
Charlotte Wales
Pierre Pignal
Max Schindler
Leonard W. Seagren
Sonja Shields
Barry Stoll
Fermin Trujillo

610.4-1990

Ronald Berlack
Susan Chonoles
William Dupras
Daniel Garvin
Shirley Gloss-Soler
John Goetz
Virl Haas
Robert Haralick
Richard Leahy
Philip Marriott
Rollin Mayer
Theo Pavlides
Azriel Rosenfeld
Leonard Seagren
Sonja Peterson Shields
Jack Sklansky
Marti Szczur
Fermin Trujillo
Terry Weymouth

610.5-1990

H. Ronald Berlack
Kathleen Briggs
Susan Chonoles
James Carlestedt
Joel Forman
Shirley A. Gloss-Soler
Jack Goetz
Virl Haas
Gina M. Little
Grace Marumoto
Anita Mires
William Malthouse
Dennis Nickle
Robert Radford
Mary Rasmussen
Roger Rohweder
Leonard Seagren
Sonja Shields
Fred Springsteel
John Sosoka
Anatol Surak
William Sutcliffe
David Usechak
Charlotte Wales
Jack Westbrook
Mary Yee

610.12-1990 [729-1983]

Russell J. Abbott
A. Frank Ackerman
Roger R. Baldwin
H. Ronald Berlack
J. David Bezek
James H. Bradley
Kathleen L. Briggs
Homer C. Carney
Susann Chonoles
Taz Daughtrey
Frank J. Douglas
William P. Dupras
John D. Earls
Mary Forcht-Tucker
David Gelperin
Al Gillen
Shirley A. Gloss-Soler
John A. Goetz
David A. Gustafson
Virl Haas
James Ingram
Gary S. Lindsay
Robert McBeth
Alicia McCurdy
Dr. José Muñoz
Geraldine Neidhart
Mary Rasmussen
Max Schindler
Paul Schmid
Leonard W. Seagren
Sonja Peterson Shields
Kevin Smith
Wayne Smith
Paul U. Thompson
Andrew H. Weigel
W. Martin Wong

Special representatives to the Computer Dictionary working group were:

Frank Jay, Christopher J. Booth, *Advisors, IEEE Standards Department*
Rollin Mayer, *Liaison, Accredited Standards Committee X3K5*

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

The sponsor for the Computer Dictionary project is the IEEE Computer Society Standards Coordinating Committee. At the time this document was published, the committee had the following membership:

Harrison Beasley
H. Ronald Berlack
William Billowitch
Richard Boberg
John Boebinger
Paul L. Borrill
Terry Bowen
Elliott Brebner
J. Reese Brown, Jr.
Lin Brown
Fletcher Buckley
Randy Bush
Clyde Camp
Steve Carter
Alan Cobb
Paul Cook
Bill Corwin
Alan Davis
Steven Deller
Bulent Dervisoglu
Bob Donnan
Paul Eastman
D. Vera Edelstein
Tim Elsmore
Dick Evans
Richard Fairley
Wayne Fischer
Kester Fong
David Gelperin
Anne Geraci
Al Gilman
John Graham
Steve Grout
Dave Gustavson
Al Hankinson
Fred Harrison
Chris Haynes
Rick Henderson
Ken Hobday
Scott Hopkinson
John Horch
Russell Housley
Charles Hudson
Marlyn Huckeby
Mike Humphrey
John Hyde
James Isaak
David James
Hal Jesperson
Richard Kalish
Matt Kaltenbach
Hans Karlsson
Freni Katki
Guy Kelley
Kim Kirkpatrick
Bob Knighten
Stanley Krolikoski
John B. Lane
Ron Leckie
Kevin Lewis
William Lidinsky
Donald C. Loughry
Al Lowenstein
Bill Maciejewski
Roger Martin
Philip Marriott
Colin Maunder
John McGrory
Louise McMonegal
Sunil Mehta
Paul Menchini
Jerry Mersky
Bennett Meyer
Louis Miller
James Mollenauer
Jim Mooney
Gary A. Nelson
Tom Pittman
Robert M. Poston
Shlomo Pri-Tal
Jane Radatz
Michael Raynham
Gordon Robinson
Larry Saunders
Richard Schmidt
Norman Schneidewind
Rudolph Schubert
David Schultz
Karen Sheaffer
Basil Sherlund
Sava Sherr
Fred Springsteel
John Starkweather
Dennis Steinauer
Robert Sulgrove
Oryal Tanir
Michael D. Teener
Donn Terry
Pat Thaler
Joseph Toy
Leonard Tripp
Margaret Updike
Eike Waltz
John W. Walz
Camille White-Partain
Les Wibberley
Cynthia Wright
John Young
Jason Zions

The following organizations supported employee participation in the development of this standard:

Apollo Computer, Inc
Applied Information Development
Atlantic Consultants
Babcock and Wilcox
BDM Corporation
Burroughs Wellcome Co.
Carney Associates
Citicorp
Computer Sciences Corporation
Computer Technology Associates, Inc.
Computron Technologies Corp.
CRS Sirrine
Datapoint Corporation
Dutchess Engineering Company
Edinboro University of Pennsylvania
Electronics Design Magazine
ETTS Inc.
Eyring Research Institute
Ford Aerospace Corporation
Frank Associates, Inc.
General Data Systems
General Electric Company
George Washington University
Georgia Institute of Technology
Harris Corporation
Hughes Aircraft Co.
IBM France
IIT Research Institute
IMI Systems
Information Spectrum, Inc.
Institute for Zero Defect Software
International Bureau of Software Test
Kansas State University
Library of Congress
Lockheed
Lockheed Saunders
Logicon, Inc.
Marine Midland Bank, N.A.
The MITRE Corporation
NASA/Goddard Space Flight Center
ORI/Intercon
Perkin-Elmer Corporation
Planning Research Corporation
Quality Assurance Institute
Rabbit Software Corp.
RCA
Sanders Associates
Sci-Tech Knowledge Systems
Siecor Corporation
SILOGIC
Softran
Space Applications Corporation
Technology Applications Group
Teledyne Brown Engineering
Televideo System, Inc.
Texas A&M University
Tonias Engineers
U.S. Air Force
U.S. Department of Housing and Urban Development
U.S. Department of the Interior
U.S. Naval Facilities
University of California at Los Angeles
University of Maryland
University of Michigan
University of Missouri
University of Southern California
University of Washington
University of Wisconsin, Madison
Wyse Technology

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

Each of the Standard Glossaries was separately balloted and approved for submission to the IEEE Standards Board. The balloting committee of each Glossary was as follows:

Balloting Committee—610.1 [1084-1986], Mathematics of Computing

A. Frank Ackerman
David A. Ackley
Leon Adams
Eleanor M. Antreassian
Geoff R. Baldwin
H. Jack Barnard
Matt Biewer
Robert M. Blasewitz
Richard Boberg
Paul L. Borrill
Kathleen L. Briggs
J. Reese Brown, Jr
Fletcher J. Buckley
David Cargo
David E. Carlson
Harold W. Carter
William C. Carter
William Cody
Christopher M. Cooke
Douglas Craft
Alan M. Davis
Robert Davis
Stephen L. Diamond
James A. Dobbins
R. A. Donnan
R. H. Douglas
Richard L. Evans
Wayne Fischer
James Flournoy
C. Rick Frederick
Carl B. Friedlander
Roger U. Fujii
Mark Garetz
David Gelperin
Anne Keller Geraci
Shirley A. Gloss-Soler
John P. Graham
Maris Graube
David B. Gustavason
Michael Patrick Hagerty
Alan L. Hankinson
Thomas L. Hannan
Paul L. Hazan
Herbert Hecht
Charles P. Hollocker
John W. Horch
William B. Humphrey
James Isaak
Richard E. James, III
Laurel V. Kaleda
Robert A. Kosman
Thomas M. Kurihara
John B. Lane
Glen G. Langdon
William P. LaPlant, Jr
William Lidinsky
Gary S. Lindsay
Donald C. Loughry
John Lowell
Philip C. Marriott
Gregg E. Marshall
Robert W. McBeth
Russell McDowell
Jerome W. Mersky
W. F. Mitchell
James F. Mollenauer
James Mooney
Perry R. Nuhn
Alice C. Parker
Thomas Pittman
John N. Postak
Robert M. Poston
Jane W. Radatz
T. D. Regulinski
John P. Riganati
Gary S. Robinson
Norman F. Schneidewind
David J. Schultz
William L. Schumacher
Leonard W. Seagren
William R. Shields
David M. Siefert
Michael Smolin
David Stevenson
James M. Syck
Richard H. Thayer
Paul U. Thompson
George D. Tice, Jr
Elias K. Tonias
Leonard L. Tripp
R. L. Van Tilburg
Eike Waltz
Richard Watters
Ronald Wazman
Lee A. Weaver
Andrew F. Wilson
Helen M. Wood
John R. Wood
John W. Young

Balloting Committee—610.2-1987, Computer Applications

Frank Ackerman
Leon Adams
Geoff Baldwin
H. Jack Barnard
H. Ronald Berlack
Matt Biewer
Robert M. Blasewitz
Richard Boberg
Paul L. Borrill
Kathleen Briggs
J. Reese Brown
Fletcher J. Buckley
David E. Carlson
Homer C. Carney
Harold W. Carter
Francois Coallier
William Cody
Christopher M. Cooke
Edward Cooper
Alan Davis
Robert Davis
Stephen L. Diamond
James A. Dobbins
R.A. Donnan
R.H. Douglas
William P. Dupras
Richard L. Evans
Dennis W. Fife
C. Rick Frederick
Carl Friedlander
Roger U. Fujii
Mark Garetz
Anne Keller Geraci
Shirley Gloss-Soler
Jack A. Goetz
Maris Graube
Virl E. Haas
Michael P. Hagerty
Alan L. Hankinson
Thomas L. Hannan
Herbert Hecht
Charles P. Hollocker
John W. Horch
William Humphrey
James Isaak
Richard James
Laurel V. Kaleda
Robert A. Kessler
Robert A. Kosman
Doug Kraft
Thomas M. Kurihara
Lah Ming Lam
John B. Lane
William P. LaPlant
William Lidinsky
John M. Long
Donald C. Loughry
John Lowell
Philip C. Marriott
Russell McDowell
Jerome W. Mersky
W. F. Mitchell
James F. Mollenauer
Geraldine R. Neidhart
Alice C. Parker
William E. Perry
Robert M. Poston
Jane Radatz
T.D. Regulinski
John P. Riganati
Norman Schneidewind
David J. Schultz
William L. Schumacher
Leonard W. Seagren
William R. Shields
David M. Siefert
Michael Smolin
David Stevenson
James M. Syck
Richard H. Thayer
Paul U. Thompson
Leonard Tripp
Richard L. Van Tillburg
Eike Waltz
John Walz
Richard Watters
Ron Waxman
Lee A. Weaver
Andrew H. Weigel
Andrew F. Wilson
W. Martin Wong
Helen M. Wood
John R. Wood

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

Balloting Committee—610.3-1989, Modeling and Simulation

David A. Ackley
Gordon Adshead
J. Aldridge
Kenneth R. Anderson
James H. Aylor
Osman Balei
Michael A. Bame
Jim Barbera
H. Ronald Berlack
Matt Biewer
Samuel J. Biondo
Kathleen L. Briggs
Jerry R. Brookshire
A. Winsor Brown
Fletcher J. Buckley
James H. Burrows
Margaret K. Butler
Harry Carl
Miguel A. Carrio
Asheem Chandna
John P. Chihorek
Frank A. Chimenti
Antonio Cicu
Francois Coallier
Christopher M. Cooke
Guy L. Copeland
Gail A. Cordes
Manuel A. D'Abreu
M. A. Daniels
Bostjan Derganc
Jim Dobbins
Frank J. Douglas
Michael Dutton
Rheodore J. Dzik
John D. Earls
Leo G.Egan
Don Eller
John W. Fendrich
Glenn S. Fields
A. M. Foley
Julian Forster
Michel Galinier
Daniel J. Garvin
Anne K. Geraci
M. Ghiassi
Sumit Ghosh
Shirley A Gloss-Soler
J.Garth Glynn
John A. Goetz
B. K. Gogia
Julio Gonzalez-Sanz
J. Kaye Grau
Andrej Grebenc
Steve Grout
L. M. Gunther
Harry E. Hansen
John D. Hallonstein
Clark M. Hay
William E. Hefley
Terry L. Hengl
John Hines
Sam Horvitz
Henry B. Hoyle
David Johnson
Walter Karplus
Peter P. Klopfenstein
Robert A. Kosman
Josep A. Krupinski
Thomas M. Kurihara
Robin B. Lake
Lak M. Lam
John B. Lane
Glen G. Langdon
Gregory N. Larsen
F. C. Lim
Deiter Look
John K. Lowell
Andy Mahindru
Henry A. Malec
Judy Malsbury
Philip C. Marriott
Roger J. Martin
Ken Massey
T. Matsubara
Robert McBeth
Paul Menchini
Jerry Mersky
Russell D. Mikel
Peter O. Mikes
Paul J. Miller
Charles S. Mooney
William A. Mullen
José Muñoz
Myron L. Nack
Saied Najafi
Matt Narotam
Geraldine R. Neidhart
Michael T. Perkins
Donald J. Pfeiffer
J.G.Phippen
John N. Postak
Jane W. Radatz
Jock A. Rader
John O. Neilson
Dennis E. Nickle
S. R. Norman
Larry O'Connell
Tuncer I. Oren
Kirit P. Parikh
Curtis H. Parks
Thomas Parrish
Max J. Schindler
Paul E. Schmid
Gregory D. Schumacher
Leonard W. Seagren
Gerard P. Shabe
Moe Shahdad
Robert W. Shillato
Ronald Short
Stanley F. Ralph
Krithivasan Ramamritham
Charles E. Rauch
Meir Razy
Thad L. D. Regulinski
Daniel Rosich
Stephen R. Schach
Victor Shtern
David M. Siefert
Lynn J. Simms
Jacob Slonim
Lyle Smith
Edward A. Straker
Oryal Tanier
Robert Thibodeau
Booker T. Thomas
Paul U. Thompson
Valentin W. Tirman
Tuyet-Lan Lydia Tran
Glendon R. Trebble
Jon Turino
Mark-Rene Uchila
Ralph Wachter
Charlotte Wales
Bruce W. Walker
John P. Walter
Paul A. Willis
Andrew F. Wilson
Paul R. Work
C. P. Wu
Alfred W. Yonda
Natalie C. Yopconka
Janusz Zalewski
Mark Zaremba
Don Zeleny

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

Balloting Committee—610.4-1990, Image Processing & Pattern Recognition

David A. Ackley
Tom Armbroster
Leo Beltracchi
Mordechai Ben-Menachem
H. Ronald Berlack
Kathleen L. Briggs
Fletcher J. Buckley
James H. Burrows
Margaret K. Butler
Harry Carl
Frank A. Chimenti
Antonio Cicu
Christopher M. Cooke
M. A. Daniels
Michael Dutton
John D. EarIs
Don Eller
John W. Fendrich
Nate Fernandez
Mehmet Ficici
Glenn S. Fields
Bruce Flinchbaugh
A. M. Foley
Julian Forster
Ismael FuentesCrespo
Leonard B. Gardner
Daniel J. Garvin
Anne K. Geraci
Shirley A Gloss-Soler
J. Garth Glynn
John A. Goetz
Julio Genzalez-Sanz
Andrej Grebenc
L. M. Gunther
Wayne Hallada
Harry E. Hansen, Jr.
Terry L. Hengl
Sam Horvitz
Frank W. James
David Johnson, III
Frank V. Jorgenson
Mansur Kabuka
Rangachar Kasturi
Yongmin Kim
Josep A. Kropinski
Thomas M. Kurihara
Lak M. Lam
John B. Lane
Glen G. Langdon, Jr.
Gregory N. Larsen
F. C. Lim
Dieter Look
Anoop Mahindru
Henry A. Malec
Judy Malsbury
Borot Maricic
Philip C. Marriott
Roger J. Martin
Tomoo Matsubara
Robert McBeth
Russell D. Mikel
Peter O. Mikes
Paul J Miller
Charles S. Mooney
R. T. Moore
Myron L. Hack
Saied Najafi
Geraldine R. Neidhart
Dennis E. Nickle
Edward A. Parrish, Jr.
Thomas Parrish
Greg Passmore
Donald J. Pfeiffer
Thomas Pittman
John N. Postak
Jane W. Radatz
Meir Razy
Thad L. D. Regulinsky
John P. Riganati
Azriel Rosenfeld
D. J. Ruest
Paul E. Schmid, III
Gregory D. Schumacher
Leonard W. Seagren
David M. Siefert
Jacob Slonim
Lyle Smith
William G. Sutcliffe
Earle E. Swartzlander
Russell E. Thiesen
Paul U. Thompson
Valentin W. Tirman, Jr.
Glendon R. Trebble
Fermin Trujillo
Mark-Rene Uchida
Robert B. Urling
W. T. van der Meere
John P. Walter
Andrew F. Wilson
Janusz Zalewski
Mark Zaremba
Don Zeleny

Balloting Committee—610.5-1990, Data Management

Michael A. Bame
Jim Barbera
H. Ronald Berlack
Kathleen L. Brims
Fletcher Buckley
Clyde Camp
Paul D. Cook
Taz Daughtrey
William P. Dupras
Anne K. Geraci
Shirley A Gloss-Soler
John A. Goetz
John W. Horch
Philip C. Marriott
Jerome W. Mersky
Dennis E. Nickle
Robert Poston
Jane Radatz
Norman Schneidewind
Robert Sulgrove
Oryal Tanir
Jon Turino

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

Balloting Committee—610.12-1990 [729], Software Engineering

Michael Bame
J. P. Barbera
Boris Beizer
Mordechai Ben-Menachem
H. Ronald Berlack
Robert Both
Kathleen L. Briggs
Winsor A. Brown
F. Buckley
Lorie J. Call
T. S. Chow
Won L. Chung
François Coallier
Michael A. Daniels
Taz Daughtrey
Daniel L. Deluna
Claudia Dencker
Bostjan K. Derganc
Merlin Dorfman
William P. Dupras
Robert E. Dwyer
Dr. L. G. Egan
Monte D. Egeland
Steven R. Eisen
John W. Fendrich
Glenn S. Fields
Jarrellann Filsinger
A. M. Foley
Thomas J. Fouser
Anne K. Geraci
Yair Gershkovitch
Edward L. Gibbs
Arthur Gittleman
Shirley A. Gloss-Soler
John A. Goetz
Julio Gonzalez Sanz
Clark Hay
Andrej Grebeno
William E. Hefley
Charles P. Hollocker
Grace M. Hopper
L. Arnold Johnson
Richard M. Karcich
Dr. Ron Kenett
Judy Kerner
R. A. Kessler
Capt. Peter P. Klopfeustein
Tom Kurihara
Lak Ming Lam
John B. Lane
Robert A. C. Lane
Gregory N. Larsen
Burton S. Liebesman
F. C. Lim
Bertil Lindberg
Harold Mains
Henry A. Malec
Philip C. Marriott
Roger J. Martin
Tomoo Matsubara
Ivano Mazzo
Glen A. Meldrum
Gene T. Morun
Hironobu Nagano
Saied Najafi
Geraldine Neidhart
Dennis E. Nickle
S. R. Norman
Tom Parrish
Michael T. Perkins
Ron Pfaff
John W. Young
Charles Plinta
John D. Pope
John N. Postak
I. C. Pyle
Jane Radatz
Gordon T. Ray
Meir Razy
John Reddon
Samuel T. Redwine
R. Waldo Roth
Steven M. Rowan
Frances Ruhlman
Dr. Stephen R. Schach
Hans Schaefer
Max Schindler
Dr. Norman Schneidewind
Wolf A. Schnoege
Greg Schumacher
R. J. Scott-Kerr
Leonard W. Seagren
Robert W. Shillato
David M. Siefert
Harry M. Sneed
Barbara J. Taute Bina
Richard H. Thayer
Terrence L. Tillmanns
Mark-Rene Uchida
Roger van Scoy
Udo Voges
Ralph Wachter
Dolores R. Wallace
John P. Walter
Andrew H. Weigel
Paul Wolfgang
Natalie C. Yopconka
Peter F. Zoll

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

Contents

SECTION PAGE

1. Scope 13

2. Dictionary Structure 13

3. Sources 13

4. Definitions 14

FIGURES PAGE

Fig 1 AND Truth Table 20
Fig 2 Anomaly 20
Fig 3 Illustrations of Arc (2) 20
Fig 4 Illustration of Background 25
Fig 5 Block Diagram 30
Fig 6 Box Diagram 32
Fig 7 B-Tree of Order 3 33
Fig 8 B′-Tree of Order 3 33
Fig 9 B*-Tree of Order 3 33
Fig 10 Bubble Chart 34
Fig 11 Call Graph 36
Fig 12 Example of Cascaded Carry 37
Fig 13 Case Construct 37
Fig 14 E is a Child Node of Node D 40
Fig 15 Illustration of Closed Curve 41
Fig 16 Illustration of Complement 45
Fig 17 Complete Binary Tree 46
Fig 18 Concatenated Key 49
Fig 19 Illustration of Concave 49
Fig 20 Illustration of Connected 52
Fig 21 Illustration of Convex 54
Fig 22 Illustration of Cross Section 56
Fig 23 Illustrations of Curve 57
Fig 24 Data Flow Diagram 60
Fig 25 Data Structure Diagram 62
Fig 26 Node E Is a Descendant Node of 66
Fig 27 Illustration of Diameter 69
Fig 28 Difference 70
Fig 29 Directed Graph 71
Fig 30 Documentation Tree 74
Fig 31 Entity/Attribute Matrix *Students* 80
Fig 32 Equivalence Truth Table 81
Fig 33 Excess-Three Code 83
Fig 34 Exclusion Truth Table 83
Fig 35 Exclusive OR Truth Table 83
Fig 36 First Normal Form 90
Fig 37 Flowchart 92
Fig 38 Graph (1) 97
Fig 39 Graph (2) 97
Fig 40 Example of a Gray Code 98

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

FIGURES PAGE

Fig 41 Heap....100
Fig 42 Hierarchy....102
Fig 43 Illustration of Hole....103
Fig 44 If-Then-Else Construct....105
Fig 45 Implication Truth Table....106
Fig 46 Input-Process-Output Chart....110
Fig 47 Intersection....114
Fig 48 Join....116
Fig 49 Multilist....135
Fig 50 NAND Truth Table....137
Fig 51 Network....138
Fig 52 NOR (1) Truth Table....140
Fig 53 NOT Truth Table....141
Fig 54 OR Truth Table....147
Fig 55 Node D Is the Parent Node for Node E....150
Fig 56 Product....159
Fig 57 Projection of Relation *Students* in Fig 11 on Attributes NAME and HOMEROOM....161
Fig 58 Qualified Name....163
Fig 59 Illustration of Region....169
Fig 60 Repeating Field....171
Fig 61 Node C Is the Root Node....175
Fig 62 Second Normal Form....178
Fig 63 Selection of Relation *Students* in Fig 11 where GRADE is ≤ 5....179
Fig 64 Nodes E and F are Sibling Nodes....182
Fig 65 Sample Software Life Cycle....186
Fig 66 Structure Chart....193
Fig 67 Graph and Subgraph....194
Fig 68 Table....197
Fig 69 Third Normal Form....202
Fig 70 Tree....206
Fig 71 Trie....206
Fig 72 Union....209
Fig 73 Example of Universal Product Code....209
Fig 74 UNTIL Construct....210
Fig 75 Example of a Venn Diagram....213
Fig 76 WHILE Construct....215

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE Standard Computer Dictionary

A Compilation of IEEE Standard Computer Glossaries

1. Scope.

This dictionary is a compilation of IEEE standard glossaries covering the fields of Mathematics of Computing, Computer Applications, Modeling and Simulation, Image Processing and Pattern Recognition, Data Management, and Software Engineering. Terms from other areas of the computer field, including Computer Graphics, Computer Networking and Data Communications, Artificial Intelligence, Computer Security and Privacy, Computer Hardware, Theory of Computation, and Computer Languages will be added in later editions.

Every effort has been made to include all terms within the designated subject areas. Terms were excluded if they were considered to be parochial to one group or organization; company proprietary or trademarked; multiword terms whose meaning could be inferred from the definitions of the component words; or terms whose meaning in the computer field could be directly inferred from their standard English meaning.

2. Dictionary Structure

Entries in the dictionary are arranged alphabetically, with the six constituent glossaries merged into a single, alphabetical presentation. An entry may consist of a single word, such as "byte," a phrase, such as "image processing," or an acronym, such as "ADP." Phrases are given in their natural order (data dictionary) rather than in reversed order (dictionary, data).

Blanks precede all other characters in alphabetizing. Hyphens and slashes are treated as blanks. Alternative spellings are shown in parentheses.

If a term has more than one definition, the definitions are numbered. In most cases, noun definitions are given first, followed by verb and adjective definitions as applicable. When a definition applies only to a particular area, such as data management, that context is indicated at the start of the definition. Examples, notes, and illustrations have been added to clarify selected definitions.

The following cross-references are used to show a term's relationship to other terms in the dictionary:

- *Contrast with* refers to a term with an opposite or substantially different meaning.
- *Syn* refers to a synonymous term.
- *See also* refers to a related term.
- *See* refers to a preferred term or to a term where the desired definition can be found.

The word "deprecated" indicates a term or definition whose use is discouraged because such use is obsolete, misleading, or ambiguous. "DoD" refers to usage by the U.S. Department of Defense.

3. Sources

The following documents are the sources for the definitions compiled into this document:

[610.1] IEEE Std 1084-1986 [610.1-1986], IEEE Standard Glossary of Mathematics of Computing Terminology.

[610.2] IEEE Std 610.2-1987, IEEE Standard Glossary of Computer Applications Terminology.

[610.3] IEEE Std 610.3-1989, IEEE Standard Glossary of Modeling and Simulation Terminology.

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052754 3

[610.4] IEEE Std 610.4-1990, IEEE Standard Glossary of Image Processing and Pattern Recognition Terminology.

[610.5] IEEE Std 610.5-1990, IEEE Standard Glossary of Data Management Terminology.

[610.12] IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology [Revision and redesignation of IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology].

The following draft standards are part of the 610 series of computer terminology glossaries. When published, they will be incorporated into later editions of this document:

[P610.6] P610.6, Glossary of Computer Graphics Terminology.

[P610.7] P610.7, Glossary of Computer Networking Terminology.

[P610.8] P610.8, Glossary of Artificial Intelligence Terminology.

[P610.9] P610.9, Glossary of Computer Security and Privacy Terminology.

[P610.10] P610.10, Glossary of Computer Hardware Terminology.

[P610.11] P610.11, Glossary of Theory of Computation Terminology.

[P610.13] P610.13, Glossary of Computer Languages.

4. Definitions

1GL. Acronym for **first generation language.** *See:* **machine language.** [610.12][1]

2GL. Acronym for **second generation language.** *See:* **assembly language.** [610.12]

3GL. Acronym for **third generation language.** *See:* **high-order language.** [610.12]

4GL. Acronym for **fourth generation language.** [610.12]

5GL. Acronym for **fifth generation language.** [610.12]

abend. Abbreviation for **abnormal end.** [610.12]

abnormal end (abend). Termination of a process prior to completion. *See also:* **abort; exception.** [610.12]

abort. To terminate a process prior to completion. *See also:* **abend; exception.** [610.12]

absolute address. An address that is permanently assigned to a device or storage location and that identifies the device or location without the need for translation or calculation. *Syn:* **explicit address; specific address.** *Contrast with:* **relative address; relocatable address; symbolic address.** *See also:* **absolute assembler; absolute code; absolute instruction; absolute loader.** [610.12]

absolute assembler. An assembler that produces absolute code. *Contrast with:* **relocating assembler.** [610.12]

absolute code. Code in which all addresses are absolute addresses. *Contrast with:* **relocatable code.** *Syn:* **specific code.** [610.12]

absolute error. (1) The amount of error expressed in the same units as the quantity containing the error. *Contrast with:* **relative error.** [610.1]
(2) Loosely, the absolute value of the error; i.e., the magnitude of the error without regard to its algebraic sign. [610.1]

absolute instruction. A computer instruction in which all addresses are absolute addresses. *See also:* **direct instruction; effective instruction; immediate instruction; indirect instruction.** [610.12]

absolute loader. A loader that reads absolute machine code into main memory, beginning at the initial address assigned to the code by the assembler or compiler, and performs no address adjustments on the code. *Contrast with:* **relocating loader.** [610.12]

[1]Numbers in brackets correspond to those of the Sources in Section 3. A number ending with the letter "a" indicates that an editorial change has been made to merge definitions from more than one source.

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

absolute value. The magnitude of a quantity without regard to its algebraic sign. [610.1]

absorbing Markov chain model. A Markov chain model that has at least one absorbing state and in which from every state it is possible to get to at least one absorbing state. [610.3]

absorbing state. In a Markov chain model, a state that cannot be left once it is entered. *Contrast with:* **nonabsorbing state.** [610.3]

abstract data type. A data type for which only the properties of the data and the operations to be performed on the data are specified, without concern for how the data will be represented or how the operations will be implemented. [610.12]

abstraction. (1) A view of an object that focuses on the information relevant to a particular purpose and ignores the remainder of the information. *See also:* **data abstraction.**[610.12]
(2) The process of formulating a view as in (1). [610.12]

abstract symbol. A symbol whose meaning and use have not been determined by a general agreement but have to be defined for each application of the symbol. [610.1]

academic simulation. *See:* **instructional simulation.** [610.3]

acceptance criteria. The criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity. *See also:* **requirement; test criteria.** [610.12]

acceptance testing. (1) Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system. [610.12]
(2) Formal testing conducted to enable a user, customer, or other authorized entity to determine whether to accept a system or component. [610.12]
Contrast with: **development testing.** *See also:* **operational testing; qualification testing.** [610.12]

access. (1) The process of obtaining data from or placing data into a storage device. *Syn:* **storage access.** *See also:* **access method; access mode.** [610.5]
(2) To obtain data from or place data into a storage device as in (1). *See also:* **direct access; indexed access; indexed sequential access; partitioned access; sequential access.** [610.5]

access method. A method for logically structuring data so that the storage location of any specific data item is well-defined. *See also:* **basic access method; direct access method; queued access method.** *Syn:* **access technique.** [610.5]

access mode. A technique that is used to access logical records within a file. *See also:* **direct access mode; file access mode; indexed sequential access mode; sequential access mode.** [610.5]

access path. The manner in which related data items are linked to one another to permit access. [610.5]

access technique. *See:* **access method.** [610.5]

access type. *See:* **file access mode.** [610.5]

accuracy. (1) A qualitative assessment of correctness, or freedom from error. [610.1, 610.12]
(2) A quantitative measure of the magnitude of error. [610.12]
Contrast with: **precision.** [610.12]

accuracy control character. A control character used to indicate whether the data with which it is associated are in error, are to be disregarded, or cannot be represented on a particular device. *Syn:* **error control character.** [610.5]

ACK. Abbreviation for **acknowledge character.** [610.5]

acknowledge character (ACK). (1) A transmission control character transmitted by a station as an affirmative response to the station with which the connection has been set up. [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

(2) A transmission control character transmitted by a receiver as an affirmative response to a sender. An acknowledge character may also be used as an accuracy control character. *See also*: **negative acknowledge character.** [610.5]

active. Pertaining to a record or file that has been accessed by one or more transactions during a given processing cycle. *See also:* **inactive; logically deleted; purged.** [610.2]

active data dictionary. A data dictionary that ensures its own consistency with a system by limiting the data items that may be used by a process to those that are defined in the data dictionary. *Syn*: **embedded data dictionary**. *Contrast with*: **passive data dictionary.** [610.5]

active file. (1) A file that is in current use. [610.5]
(2) A file with an expiration date that has not yet been reached. [610.5]

active redundancy. In fault tolerance, the use of redundant elements operating simultaneously to prevent, or permit recovery from, failures. *Contrast with:* **standby redundancy.** [610.12]

activity. In modeling and simulation, a task that consumes time and resources and whose performance is necessary for a system to move from one event to the next.[610.3]

activity-based simulation. A discrete simulation that represents the components of a system as they proceed from activity to activity; for example, a simulation in which a manufactured product moves from station to station in an assembly line. [610.3]

activity ratio. The ratio of active records to the total number of records in a file. [610.2]

actual instruction.* *See:* **effective instruction.**
* Deprecated. [610.12]

actual key. A key that directly expresses the physical location of a logical record on a storage medium. [610.5]

actual parameter. *See:* **argument (3).** [610.12]

acutance. A measure of the sharpness of the edges in an image. [610.4]

ad hoc (adhoc). Pertaining to an item such as a computer program or database used for a particular and specific purpose; for example, an ad hoc query. *Note*: Usually the item is used for a relatively short time, then discarded. [610.5]

ad hoc query. A query that is used for a particular and specific purpose. *Note*: Such a query is usually used once or twice, then discarded. [610.5]

Ada. *Note:* Ada and other specific computer languages are defined in P610.13. [610.12, P610.13]

adaptability. *See:* **flexibility.** [610.12]

adaptation data. Data used to adapt a program to a given installation site or to given conditions in its operational environment. [610.12]

adaptation parameter. A variable that is given a specific value to adapt a program to a given installation site or to given conditions in its operational environment; for example, the variable Installation_Site_Latitude. [610.12]

adaptive coding. The application of two or more image compression techniques to a single image, based on properties of different parts of the image. [610.4]

adaptive maintenance. Software maintenance performed to make a computer program usable in a changed environment. *Contrast with:* **corrective maintenance; perfective maintenance.** [610.12]

add. To insert a record into an existing file. [610.5]

add file. A file containing records that are being added or are to be added to a master file. [610.5]

add record. A record that is to be added or which has been added to a master file. *Contrast with*: **deletion record.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

add transaction. A transaction that causes a new record to be added to a master file. *See also:* **change transaction; delete transaction; null transaction; update transaction.** [610.2]

addend. A number to be added to another number (the augend) to produce a result (the sum). [610.1]

addition without carry.* *See:* **exclusive OR.**
*Deprecated. [610.1]

address. (1) A number, character, or group of characters that identifies a given device or storage location. [610.12]
(2) To refer to a device or storage location by an identifying number, character, or group of characters. [610.12]
See also: **absolute address; effective address; implied addressing; indirect address; relative address; relocatable address; symbolic address; virtual address.** [610.12]

address calculation sort. An insertion sort in which each of the items to be sorted is inserted into one of several lists, according to an address calculated from its value, and the resulting lists are then merged. *Syn:* **multiple list insertion sort.** [610.5]

address field. Any of the fields of a computer instruction that contain addresses, information necessary to derive addresses, or values of operands. *Syn:* **address part.** *Contrast with:* **operation field.** [610.12]

address format. (1) The number and arrangement of address fields in a computer instruction. *See also:* ***n*-address instruction; *n*-plus-one address instruction.** [610.12]
(2) The number and arrangement of elements within an address, such as the elements needed to identify a particular channel, device, disk sector, and record in magnetic disk storage. [610.12]

addressing exception. An exception that occurs when a program calculates an address outside the bounds of the storage available to it. *See also:* **data exception; operation exception; overflow exception; protection exception; underflow exception.** [610.12]

address modification. Any arithmetic, logical, or syntactic operation performed on an address. *See also:* **effective address; indexed address; relative address; relocatable address.** [610.12]

address part. *See:* **address field.** [610.12]

address space. (1) The addresses that a computer program can access. *Note:* In some systems, this may be the set of physical storage locations that a program can access, disjoint from other programs, together with the set of virtual addresses referring to those storage locations, which may be accessible by other programs. [610.12]
(2) The number of memory locations that a central processing unit can address. [610.12]

address table sorting. A sorting technique in which a table of addresses that point to the items to be sorted is manipulated instead of moving the items themselves. *See also:* **key sorting; list sorting.** [610.5]

Adel'son-Velskii and Landis (AVL) tree. A height-balanced binary tree in which the difference in height of the two subtrees of any node is at most 1. *Note:* Also referred to as a HB tree; a height-balanced 1-tree. [610.5]

adjacency. In character recognition, a condition in which the character spacing reference lines of two consecutive characters printed on the same line are separated by less than a specified distance. [610.2]

adjust line mode. In text formatting, an operating mode in which line endings are automatically adjusted to comply with the current margin setting. *See also:* **text end adjustment; word wrap.** [610.2]

administrative data processing (ADP). The use of computers for administrative applications such as personnel, payroll, and accounting functions. [610.2]

ADP. (1) Acronym for automatic data processing; automated data processing. [610.2]
(2) Acronym for administrative data processing. [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

afferent. Pertaining to a flow of data or control from a subordinate module to a superordinate module in a software system. *Contrast with:* **efferent.** [610.12]

aggregate. *See:* **data aggregate.** [610.5]

AKWIC. Acronym for **author and keyword in context.** *See:* **author and keyword in context index.** [610.2]

algebraic coding function. In hashing, a hash function that returns the result of evaluating some polynomial in which selected digits of the original key are used as coefficients. For example, in the function below, the first three digits of the original key are evaluated as a, b, and c, respectively, in the polynomial $a + b\,x + c\,x^2$ with x=14. [610.5]

Original key	Calculation	Hash value
964721	$9 + 6(14) + 4(14)^2 = 877$	877
864765	$8 + 6(14) + 4(14)^2 = 876$	876

algebraic language. A programming language that permits the construction of statements resembling algebraic expressions, such as $Y = X + 5$. For example, FORTRAN. *See also:* **algorithmic language; list processing language; logic programming language.** [610.12]

algebraic manipulation. The processing of mathematical expressions without concern for the numeric values of the symbols that represent numbers. [610.1]

algorithm. (1) A finite set of well-defined rules for the solution of a problem in a finite number of steps; for example, a complete specification of a sequence of arithmetic operations for evaluating sine x to a given precision. [610.1, 610.12]
(2) Any sequence of operations for performing a specific task. [610.12]

algorithmic language. A programming language designed for expressing algorithms; for example, ALGOL. *See also:* **algebraic language; list processing language; logic programming language.** [610.12]

alias. *See:* **alternate name.** [610.5]

alignment. Pertaining to data that are stored beginning at certain machine-dependent boundaries. Such data is said to be "aligned," otherwise it is said to be "unaligned;" for example, a four-bit data item is aligned if it begins on a full-word boundary of eight-bit words. *Syn:* **boundary alignment.** [610.5]

allocated baseline. In configuration management, the initial approved specifications governing the development of configuration items that are part of a higher level configuration item. *Contrast with:* **developmental configuration; functional baseline; product baseline.** *See also:* **allocated configuration identification.** [610.12]

allocated configuration identification. In configuration management, the current approved specifications governing the development of configuration items that are part of a higher level configuration item. Each specification defines the functional characteristics that are allocated from those of the higher level configuration item, establishes the tests required to demonstrate achievement of its allocated functional characteristics, delineates necessary interface requirements with other associated configuration items, and establishes design constraints, if any. *Contrast with:* **functional configuration identification; product configuration identification.** *See also:* **allocated baseline.** [610.12]

allocation. (1) The process of distributing requirements, resources, or other entities among the components of a system or program. [610.12]
(2) The result of the distribution in (1). [610.12]

ALP. Acronym for **automated language processing.** [610.2]

alpha. (1) Abbreviation for **alphabetic.** [610.5]
(2) * Abbreviation for **alphanumeric.** [610.5]
* Deprecated.

alphabet. An ordered set of all the letters or symbols used in a language, including letters with diacritical signs where appropriate, but not including punctuation marks. [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

alphabetic. Pertaining to data that consist solely of letters from the same alphabet. For example, (AaBbCcDdEe...) plus the space character. *Syn*: **alpha.** *See also*: **alphanumeric; character.** [610.5]

alphabetic character set. A character set that contains alphabetic characters and that may contain control characters, special characters, and the space character, but not digits. [610.5]

alphabetic code. A code that uses alphabetic characters to represent data. [610.5]

alphabetic shift. A control for selecting the alphabetic character set on an keyboard or printer. *Contrast with*: **numeric shift.** *See also*: **shift character.** [610.5]

alphabetic string. A character string consisting solely of alphabetic characters. [610.5]

alphameric. Abbreviation for **alphanumeric.** [610.5]

alphanumeric (alphameric). Pertaining to data that contain the letters of an alphabet (AaBbCcDdEeFfGgHh...), the decimal digits (0123456789), and may contain control characters, special characters and the space character. [610.5]

alphanumeric character set. A character set that contains alphanumeric characters. [610.5]

alphanumeric code. A code that uses alphanumeric characters to represent data. [610.5]

alter. (1) To insert, delete, or modify a data record. [610.5]
(2) To change a logical relationship or physical structure of a database. *See also*: **modify.** [610.5]

alternate index. An index (2) that uses alternate keys to reference indexed data. *See also*: **secondary index.** [610.5]

alternate key. (1) In a relation, a candidate key that is not chosen to be the primary key for that relation. [610.5]
(2) A secondary key for an indexed sequential file. *See also*: **alternate index; prime key.** [610.5]

alternate name. Any name besides the data element name by which a data item is known. *Note*: Often stored in data dictionaries. *Syn*: **alias.** [610.5]

American National Standard Code for Information Interchange (ASCII). A binary code in which 128 letters, numbers, and special characters are represented by seven-bit numerals. *Note:* Some systems make use of an eight-bit binary code, called ASCII-8, in which 256 symbols are represented. [610.1]

amphiboly. A logical expression that has more than one meaning. For example, the expression A OR B AND C might mean (A OR B) AND C or A OR (B AND C) depending upon the rules of interpretation used. [610.1]

analog. Pertaining to data in the form of continuously variable physical quantities. *Contrast with:* **digital.** [610.1]

analog representation. The representation of numerical quantities by means of continuous physical variables such as translation, rotation, voltage, or resistance. *Contrast with:* **digital representation.** [610.1]

analog simulation. (1) A simulation that is designed to be executed on an analog system. [610.3]
(2) A simulation that is designed to be executed on a digital system but that represents an analog system. [610.3]
(3) A simulation of an analog circuit. *Contrast with:* **digital simulation.** *See also:* **hybrid simulation.** [610.3]

analytical model. A model consisting of a set of solvable equations; for example, a system of solvable equations that represents the laws of supply and demand in the world market. [610.3]

ancestor. Relative to a given node x within a tree, any node y for which x is a descendent node of y. [610.5]

AND. A Boolean operator having the property that if P is a statement, Q is a statement, R is

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

a statement,..., then the AND of P,Q,R,... is true if and only if all statements are true. *Note:* P AND Q is often represented by P·Q, P&Q, P∧Q, or PQ. *Syn:* **Boolean multiplication; collation; conjunction; intersection; logic multiply; logical multiply; meet.**[610.1]

P	Q	P ∧ Q
0	0	0
0	1	0
1	0	0
1	1	1

Fig 1
AND Truth Table

AND-NOT.* *See:* **exclusion.** [610.1]
*Deprecated.

annual cycle. One complete execution of a data processing function that must be performed once a year. *See also:* **daily cycle; monthly cycle; quarterly cycle; weekly cycle.** *Syn:* **yearly cycle.** [610.2]

anomaly. (1) In software engineering anything observed in the documentation or operation of software that deviates from expectations based on previously verified software products or reference documents. [610.12a] (2) In data management, an irregularity that arises when processing an improperly structured database. For example, in order to retrieve all the SUPPLIERS from the database in Fig 2, one would have to search sequentially through all the PARTS INVENTORY segments. This is said to be an *anomaly.* [610.5a]

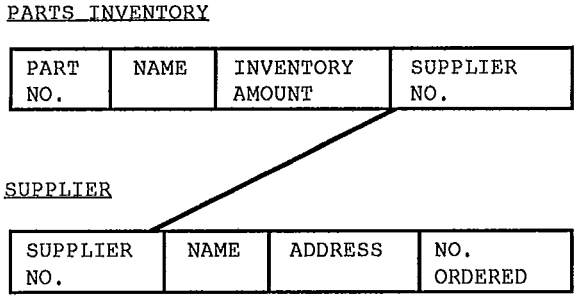


Fig 2
Anomaly

anticipatory buffering. A buffering technique in which data are stored in a buffer in anticipation of a need for the data. *See also:* **dynamic buffering; simple buffering.** [610.12]

anticipatory paging. A storage allocation technique in which pages are transferred from auxiliary storage to main storage in anticipation of a need for those pages. *Contrast with:* **demand paging.** [610.12]

application. The use to which a computer system is put; for example, a payroll application, an airline application, or a network application. [610.2, 610.5]

application generator. A code generator that produces programs to solve one or more problems in a particular application area; for example, a payroll generator. [610.12]

application-oriented language. A computer language with facilities or notations applicable primarily to a single application area; for example, a language for computer-assisted instruction or hardware design. *See also:* **authoring language; specification language; query language.** [610.12]

application program. A computer program that is used for a specific application. [610.5]

application software. Software designed to fulfill specific needs of a user; for example, software for navigation, payroll, or process control. *Contrast with:* **support software; system software.** [610.12]

application view. *See:* **logical database.** [610.5]

APT. Acronym for **automatically programmed tools.** [610.2]

arc. (1) A continuous portion of a circle. [610.4] (2) A finite set of pixels representing a portion of a curve. [610.4]



Fig 3
Illustrations of Arc (2)

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

architectural design. (1) The process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system. *See also:* **functional design.** [610.12]
(2) The result of the process in (1). [610.12]

architecture. The organizational structure of a system or component. *See also:* **component; module; subprogram; routine.** [610.12]

archival database. A copy of a database saved for later reference or use. [610.5]

area. (1) A named collection of records within a database. *Note*: May contain occurrences of one or more record types, and a record type may have occurrences in one or more area. *Syn*: **realm.** [610.5]
(2) In CODASYL, a part of a database that can be opened or closed as a unit. *Note*: This term was used in early CODASYL documents, but is now considered deprecated. [610.5]
(3) In image processing, the number of pixels in a region. [610.4]

argument. (1) An independent variable; for example, the variable m in the equation $E = mc^2$. [610.12]
(2) A specific value of an independent variable; for example, the value $m = 24$ kg. [610.12]
(3) A constant, variable, or expression used in a call to a software module to specify data or program elements to be passed to that module. *Syn:* **actual parameter.** *Contrast with:* **formal parameter.** [610.12]

arithmetic. Pertaining to data that has the characteristics of base, scale, mode, and precision. *Note*: Used to represent numbers. *See also:* **binary picture data; coded arithmetic data; decimal picture data; numeric data.** *Contrast with*: **string.** [610.5]

arithmetic check. *See:* **mathematical check.** [610.1]

arithmetic expression. An expression containing any combination of variables and constants joined by one or more arithmetic operators such that the expression can be reduced to a single numerical result. [610.1]

arithmetic operation. An operation that is performed in accordance with the rules of ordinary arithmetic. [610.1]

arithmetic overflow. *See:* **overflow.** [610.1]

arithmetic point. *See:* **radix point.** [610.1]

arithmetic shift. A shift that affects all digit positions in a register, word, or numeral but does not affect the sign position. For example, +231.702 shifted two places to the left becomes + 170.200. *Note:* The result is equivalent to multiplication or division by an integral power of the radix, except for the truncation effects. *Syn:* **numerical shift.** *Contrast with:* **logical shift.** [610.1]

arithmetic underflow. *See:* **underflow.** [610.1]

array. An n-dimensional ordered set of data items identified by a single name and one or more indices, so that each element of the set is individually addressable. *See also:* **linear array.** [610.5, 610.12]

artificial intelligence. *Note:* P610.8 will define terminology pertaining to artificial intelligence. [610.12]

artificial language. *See:* **formal language.** [610.12]

ascender. The portion of a graphic character that extends above the main part of the character; for example, the upper portion of the letters b and h. *Contrast with:* **descender.** [610.2]

ASCII. Acronym for **American National Standard Code for Information Interchange.** [610.1]

assemble. To translate a computer program expressed in an assembly language into its machine language equivalent. *Contrast with:* **compile; disassemble; interpret.** [610.12]

assemble-and-go. An operating technique in which there are no stops between the assembling, linking, loading, and execution of a computer program. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

assembled origin. The address of the initial storage location assigned to a computer program by an assembler, a compiler, or a linkage editor. *Contrast with:* **loaded origin.** *See also:* **offset (1); starting address.** [610.12]

assembler. A computer program that translates programs expressed in assembly language into their machine language equivalents. *See also:* **absolute assembler; cross-assembler; relocating assembler.** *Contrast with:* **compiler; interpreter.** [610.12]

assembler code. *See:* **assembly code.** [610.12]

assembler language. *See:* **assembly language.** [610.12]

assembly code. Computer instructions and data definitions expressed in a form that can be recognized and processed by an assembler. *Syn:* **assembler code.** *Contrast with:* **compiler code; interpretive code; machine code.** [610.12]

assembly language. A programming language that corresponds closely to the instruction set of a given computer, allows symbolic naming of operations and addresses, and usually results in a one-to-one translation of program instructions into machine instructions. *Syn:* **assembler language; low level language; second generation language.** *Contrast with:* **fifth generation language; fourth generation language; high-order language; machine language.** *Note:* Specific languages will be defined in P610.13. [610.12, P610.13]

assertion. A logical expression specifying a program state that must exist or a set of conditions that program variables must satisfy at a particular point during program execution. Types include input assertion, loop assertion, output assertion. *See also:* **invariant; proof of correctness.** [610.12]

assigned indexing. Automatic indexing in which appropriate keywords are assigned from a list of preselected keywords rather than from the text of the document or information being indexed. *Syn:* **assignment indexing.** *Contrast with:* **derivative indexing.** [610.2]

assignment indexing. *See:* **assigned indexing.** [610.2]

assignment statement. A computer program statement that assigns a value to a variable; for example, $Y := X - 5$. *Contrast with:* **control statement; declaration.** *See also:* **clear; initialize; reset.** [610.12]

association. In data management, a relationship established in a data model to represent a connection between entities that is not reflected solely by the attributes inherent in the entities. [610.5]

associative lookup. Table lookup performed on a table that is stored in associative memory. [610.5]

assumed binary point. The position in a binary numeral at which the binary point is assumed to be located; usually at the right unless otherwise specified. *Syn:* **implied binary point.** [610.1]

assumed decimal point. The position in a decimal numeral at which the decimal point is assumed to be located; usually at the right unless otherwise specified. *Syn:* **implied decimal point.** [610.1]

assumed radix point. The position in a numeral at which the radix point is assumed to be located; usually at the right unless otherwise specified. *Syn:* **implied radix point.** [610.1]

ATM. Acronym for **automated teller machine.** [610.2]

atomic condition. The basic qualification condition in a query, consisting of the name of a data item, a logical operation, and a value; for example, LASTNAME = 'Jones.' *See also:* **item condition; query condition; record condition.** [610.5]

atomic data element. A data element that cannot be broken into constituent data elements. *Contrast with:* **composite data element.** [610.5]

atomic type. A data type, each of whose members consists of a single, nondecomposable

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

data item. *Syn:* **primitive type.** *Contrast with:* **composite type.** [610.12]

attribute. (1) A characteristic of an item; for example, the item's color, size, or type. *See also*: **data attribute; quality attribute.** [610.5, 610.12]
(2) In a relation, a named characteristic, property, or description of an entity. *Note*: Also known as data element, data field, data item or column in a table. *See also*: **column.** *Syn*: **data field.** [610.5]

attribute data element. A data element within a record that represents a property, feature, or characteristic of the subject of that record; for example, the data element "date of birth" in a record containing "name," "address," and "date of birth" of a person. *Contrast with*: **primary data element.** [610.5]

audit. An independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria. *See also:* **functional configuration audit; physical configuration audit.** [610.12]

audit trail. A manual or computerized record that can be used to trace the transactions affecting the contents of a record or a file.[610.2]

augend. A number to which another number (the addend) is added to produce a result (the sum). [610.1]

author and keyword in context (AKWIC) index. A variation of a keyword in context (KWIC) index in which author and keyword entries are combined and presented in a KWIC format. *Contrast with:* **word and author index.** [610.2]

authoring language. A high level programming language used to develop courseware for computer-assisted instruction. *See also:* **authoring system.** [610.2, 610.12]

authoring system. A programming system that incorporates an authoring language. [610.2, 610.12]

automated data medium. *See*: **machine-readable medium.** [610.5]

automated data processing (ADP). *See:* **automatic data processing.** [610.2]

automated dictionary. In machine-aided translation, an automated lexicon in which entries are single words. *Contrast with:* **automated glossary.** [610.2]

automated glossary. In machine-aided translation, an automated lexicon in which entries may consist of multiple words. *Contrast with:* **automated dictionary.** *Syn:* **terminology bank.** [610.2]

automated language processing (ALP). The application of data processing, word processing, and machine-aided translation to the processing or translation of natural languages. [610.2]

automated lexicon. A computer-resident table of source language and target language equivalents that serves as the central component in a machine-aided translation system. *See also:* **automated dictionary; automated glossary.** [610.2]

automated office. *See:* **electronic office.** [610.2]

automated teller machine (ATM). An unattended terminal-type device that offers simple banking services such as cash withdrawals, transfer of funds between accounts, and account balance inquiry. *Syn:* **customer bank communications terminal.** [610.2]

automated thesaurus. In machine-aided translation, a computer-resident thesaurus used in conjunction with an automated lexicon to handle words with multiple meanings. [610.2]

automated verification system. (1) A software tool that accepts as input a computer program and a representation of its specification and produces, possibly with human help, a proof or disproof of the correctness of the program.
(2) Any software tool that automates part or all of the verification process. [610.12]

automatic. Pertaining to a function, operation, process, or device that, under specified

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

conditions, functions without intervention by a human operator. [610.2]

automatic abstracting. In library automation, the automatic selection of words and phrases from a document to produce an abstract. [610.2]

automatically programmed tools (APT). A programming system using English-like symbolic descriptions of part and tool geometry and tool motion for numerical control. [610.2]

automatic calendar. A component of some office automation systems that allows users to store their appointments in a database and to set up meetings by requesting a search for an available meeting time in each of the participants' calendars. [610.2]

automatic check. A check that is built into a device in order to verify the accuracy of information transmitted, manipulated, or stored by that device. *Syn:* **built-in check; hardware check.** [610.5]

automatic control. Describes a control system capable of operating without external or human intervention. *See also:* **numerical control; process control.** [610.2]

automatic data processing (ADP). Data processing performed by a computer system. *Syn:* **automated data processing; electronic data processing.** [610.2]

automatic hyphenation. In text formatting, hyphenation in which all line-ending and word break decisions are made automatically. Word break decisions may be made using syllabication algorithms or a dictionary containing commonly used words and their syllables. *See also:* **manual hyphenation; semi-manual hyphenation.** [610.2]

automatic index. An index produced by automatic indexing. *See also:* **keyword in context index; keyword out of context index; permutation index; selective listing in combination index; word index.** [610.2]

automatic indexing. Automated production of an index by selecting keywords and organizing them according to the type of index being produced. *Note:* Methods include assigned indexing and derivative indexing. *See also:* **keyword in context index; keyword out of context index; permutation index; selective listing in combination index; word index.** [610.2]

automatic pagination. In text formatting, the automatic arrangement or rearrangement of text according to preset page layout parameters such as margin width and lines per page. *Note:* May also include the assignment and placement of page numbers on the pages. *Syn:* **pagination; repagination.** [610.2]

automation. (1) The implementation of a process by automatic means. [610.2]
(2) The theory, art, or technique of making a process more automatic. [610.2]
(3) The investigation, design, development, and application of methods of rendering processes automatic, self-moving, or self-controlling. [610.2]

availability. The degree to which a system or component is operational and accessible when required for use. Often expressed as a probability. *See also:* **error tolerance; fault tolerance; robustness.** [610.12]

AVL tree. Acronym for Adel'son-Velskii and Landis tree. A height-balanced binary tree. *Note*: Also referred to as a HB tree; a height balanced 1-tree. [610.5]

back out. *See*: **rollback.** [610.5]

back-to-back testing. Testing in which two or more variants of a program are executed with the same inputs, the outputs are compared, and errors are analyzed in case of discrepancies. *See also:* **mutation testing.** [610.12]

background. (1) In job scheduling, the computing environment in which low-priority processes or those not requiring user interaction are executed. *Contrast with:* **foreground.** *See also:* **background processing.** [610.12]
(2) In micrographics, the portion of a

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

document that does not contain lettering or other information. [610.2]
(3) In image processing, a connected component of a region's complement such that the connected component completely surrounds the region. [610.4]

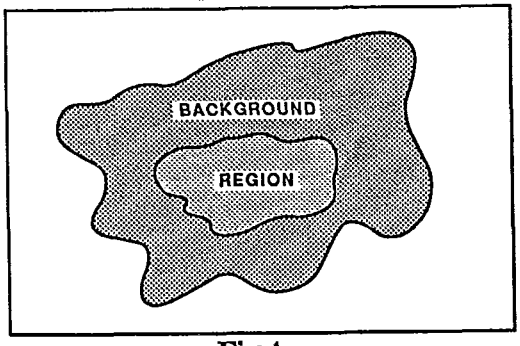


Fig 4
Illustration of Background

background processing. The execution of a low-priority process while higher priority processes are not using computer resources, or the execution of processes that do not require user interaction. *Contrast with:* **foreground processing.** [610.12]

backspace character (BS). A format effector character that causes the print or display position to move one position backward along the line without producing the printing or display of any graphic. [610.5]

backup (back-up). (1) A system, component, file, procedure, or person available to replace or help restore a primary item in the event of a failure or externally-caused disaster. [610.5, 610.12]
(2) To create or designate a system, component, file, procedure, or person as in (1). [610.5, 610.12]

backup programmer. The assistant leader of a chief programmer team; responsibilities include contributing significant portions of the software being developed by the team, aiding the chief programmer in reviewing the work of other team members, substituting for the chief programmer when necessary, and having an overall technical understanding of the software being developed. *See also:* **chief programmer.** [610.12]

backward execution. *See:* **reversible execution.** [610.12]

backward recovery. (1) The reconstruction of a file to a given state by reversing all changes made to the file since it was in that state. *Contrast with:* **forward recovery; inline recovery.** [610.5, 610.12]
(2) A type of recovery in which a system, program, database, or other system resource is restored to a previous state in which it can perform required functions. [610.12]

balanced error. (1) A set of error values in which the maximum and minimum are opposite in sign and equal in magnitude. [610.1]
(2) A set of error values whose average is zero. *Contrast with:* **unbalanced error.** [610.1]

balanced merge. A merge in which the subsets to be merged are equally distributed among half of the available storage, then the subsets are merged onto the other half of storage. *Contrast with:* **unbalanced merge.** [610.5]

balanced merge sort. A merge sort in which the sorted subsets created by internal sorts are equally distributed among half of the available storage, the subsets are merged onto the other half of the available storage, and this process is repeated until all the items are in one sorted set. *Contrast with:* **unbalanced merge sort.** [610.5]

balanced tree. *See:* **height-balanced tree.** [610.5]

BAM. Acronym for **basic access method.** [610.5]

bar code. An identification code consisting of a pattern of vertical bars whose width and spacing identifies the item marked. *Note:* The code is meant to be read by an optical input device, such as a bar code scanner. Applications include retail product pricing labels, identification of library documents, and railroad box car identification. *See also:* **universal product code.** *Syn:* **optical bar code.** [610.2]

base. *See:* **radix.** [610.1]

base address. An address used as a reference point to which a relative address is added to

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052766 T

determine the address of the storage location to be accessed. *See also:* **indexed address; relative address; self-relative address.** [610.12]

base complement. *See:* **radix complement.** [610.1]

baseline. (1) A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures. [610.12]
(2) A document or a set of such documents formally designated and fixed at a specific time during the life cycle of a configuration item. *Note:* Baselines, plus approved changes from those baselines, constitute the current configuration identification. *See also:* **allocated baseline; developmental configuration; functional baseline; product baseline.** [610.12]
(3) Any agreement or result designated and fixed at a given time, from which changes require justification and approval. [610.12]

baseline management. In configuration management, the application of technical and administrative direction to designate the documents and changes to those documents that formally identify and establish baselines at specific times during the life cycle of a configuration item. [610.12]

base-minus-one complement. *See:* **diminished-radix complement.** [610.1]

base notation. *See:* **radix notation.** [610.1]

base number. *See:* **radix.** [610.1]

base point. *See:* **radix point.** [610.1]

base relation. A relation that is not derivable from other base relations in a given database. *Contrast with*: **derived relation.** [610.5]

basic access method (BAM). An access method in which each input or output statement invokes a corresponding machine operation. For example, when reading a file with 10 records, exactly 10 READ operations will be invoked. *Contrast with*: **direct access method; queued access method.** *See also*: **basic direct access method; basic indexed sequential access method; basic partitioned access method; basic sequential access method.** [610.5]

basic direct access method (BDAM). A variation on the basic access method that allows direct access to the data. [610.5]

basic indexed sequential access method (BISAM). A variation on the basic access method that allows indexed sequential access to the data. *See also*: **queued indexed sequential access method.** [610.5]

basic partitioned access method (BPAM). A variation on the basic access method that allows partitioned access to the data. [610.5]

basic sequential access method (BSAM). A variation of the basic access method that allows sequential access to the data. *See also*: **indexed sequential access method; queued sequential access method; virtual sequential access method.** [610.5]

batch. Pertaining to a system or mode of operation in which inputs are collected and processed all at one time, rather than being processed as they arrive, and a job, once started, proceeds to completion without additional input or user interaction. *Contrast with:* **conversational; interactive; on-line; real time.** [610.12]

Batcher's parallel sort. A merge sort in which corresponding items in two ordered subsets are simultaneously compared and, if necessary, exchanged; the resulting subsets are divided in half and interleaved with one another, and these steps are repeated until the merge is complete. *Note*: This algorithm is particularly appropriate for parallel processing. *Syn*: **merge exchange sort; odd-even sort.** *See also*: **bitonic sort.** [610.5]

bathtub curve. A graph of the number of failures in a system or component as a function of time. The name is derived from the usual shape of the graph: a period of decreasing failures (the early-failure period), followed by a relatively steady period (the constant-failure period), followed by a period of

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

increasing failures (the wearout-failure period). [610.12]

BCD. Acronym for **binary-coded decimal.** [610.1]

BCD real data. Acronym for **binary-coded decimal real data.** [610.5]

BCNF. Acronym for **Boyce/Codd normal form.** [610.5]

BDAM. Acronym for **basic direct access method.** [610.5]

BDP. Acronym for **business data processing.** [610.2]

behavioral model. *See:* **black box model.** [610.3]

BEL. Abbreviation for **bell character.** [610.5]

bell character (BEL). A control character that is used when there is a need to call for human attention and that may activate an alarm or other attention devices. [610.5]

benchmark. (1) A standard against which measurements or comparisons can be made. [610.12]
(2) A procedure, problem, or test that can be used to compare systems or components to each other or to a standard as in (1). [610.12]
(3) A recovery file. [610.12]

bias. (1) A systematic deviation of a value from a reference value. [610.1]
(2) The amount by which the average of a set of values departs from a reference value. *Syn:* **bias error.** [610.1]

biased exponent. In floating-point arithmetic, the sum of the exponent and a constant (bias) chosen to make the biased exponent's range nonnegative. [610.1]

bias error. *See:* **bias (2).** [610.1]

big-bang testing. A type of integration testing in which software elements, hardware elements, or both are combined all at once into an overall system, rather than in stages. [610.12]

binary. (1) Pertaining to a selection in which there are two possible outcomes. [610.1, 610.5]
(2) Pertaining to the numeration system with a radix of two. *Syn:* **natural binary; normal binary; ordinary binary; pure binary; regular binary; standard binary; straight binary.** [610.1, 610.5]

binary arithmetic operation. An arithmetic operation in which the operands and the results are represented in the binary numeration system. [610.1]

binary Boolean operation.* *See:* **dyadic Boolean operation.** [610.1]
*Deprecated.

binary chop. * *See*: **binary search.** [610.5]
* Deprecated.

binary code. A code that uses exactly two distinct characters, usually 0 and 1, to represent data or instructions. [610.1, 610.5]

binary-coded decimal (BCD). Pertaining to a number representation system in which each decimal digit is represented by a unique arrangement of binary digits (usually four); for example, the number 23 is represented as 0010 0011, whereas in binary notation, 23 is represented as 10111. *Syn:* **coded decimal.** [610.1]

binary coded decimal character set. A character set containing all 64 characters that can be represented as permutations of six bits. [610.5]

binary coded decimal (BCD) real data. A technique for assigning numeric characters such that each decimal digit is represented by a unique arrangement of binary digits with an implied radix point at a specified position. [610.5]

decimal	163.3_{10}
BCD real	$0001\ 0110\ 0011\ .\ 0011_2$

binary-coded digit. A digit of any number representation system that is represented as a fixed number of binary digits. For example, the decimal digit 9 is represented as 1001. [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

binary-coded octal. Pertaining to a three-bit binary code in which the octal digits 0–7 are represented by the binary numerals 000-111. [610.1]

binary data. Numeric data used to represent binary digits. *See also*: **binary picture data; fixed-point binary data; packed binary data.** [610.5]

binary digit (bit). (1) A unit of information that can be represented by either a zero or a one.
(2) An element of computer storage that can hold a unit of information as in (1).
(3) A numeral used to represent one of the two digits in the binary numeration system; zero (0) or one (1).
See also: **binary element; byte; word.** [610.1a, 610.5, 610.12]

binary digit character. A character within a picture specification that represents a binary digit. [610.5]

binary element. A data element that can assume either of two possible values or states. *See also*: **binary digit, binary variable.** [610.1, 610.5]

binary element string. A string consisting solely of binary elements. [610.1]

binary image. A digital image in which each pixel is assigned a value of either zero or one. [610.4]

binary incremental representation. An incremental representation system in which the value of an increment is plus one or minus one. *Syn:* **incremental binary representation.** [610.1]

binary insertion sort. An insertion sort in which each item in the set to be sorted is inserted into its proper position in the sorted set using a binary search algorithm. *Contrast with*: **two-way insertion sort.** [610.5]

binary notation. Any notation that uses the binary digits and the radix 2. *Syn:* **binary scale; two-scale.** [610.1]

binary number. (1) A quantity that is expressed by using the binary numeration system. [610.1]
(2) Loosely, a binary numeral. [610.1]

binary number system. *See:* **binary numeration system.** [610.1]
*Deprecated.

binary numeral. A numeral in the binary numeration system. For example, the binary numeral 101 is equivalent to the decimal numeral 5. [610.1]

binary numeration system. The numeration system that uses the binary digits and the radix 2. *Syn:* **binary system; pure binary numeration system.** [610.1]

binary one. The "true" binary state, usually represented as 1 or T. *Contrast with:* **binary zero.** [610.1]

binary operation.* (1) *See:* **Boolean operation.**
(2) *See:* **dyadic operation.** [610.1]
*Deprecated.

binary operator. *See:* **dyadic operator.** [610.1]

binary picture data. Arithmetic data that is associated with a picture specification that allows binary digit characters, a radix point, exponent characters, and sign characters. *Syn*: **numeric bit data.** *Contrast with*: **decimal picture data.** [610.5]

binary point. The radix point in the binary numeration system. [610.1]

binary radix trie search. A radix trie search using a binary trie in which only one bit is considered on each branch. *See also*: **multiway radix trie search.** [610.5]

binary relation. A relation with two attributes. [610.5]

binary scale. *See:* **binary notation.** [610.1]

binary search. A dichotomizing search in which, at each step of the search, the remaining set of items is partitioned into two equal parts. *Syn*: **binary chop; bisection; logarithmic search.** *Contrast with*: **fibonacci search; interpolation search.** *See also:* **bi-**

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

nary search tree; binary tree search. [610.5]

binary search tree. A search tree of order 2. [610.5]

binary system. *See:* **binary numeration system.** [610.1]

binary-state variable. *See:* **binary variable.** [610.1]

binary-to-decimal conversion. The process of converting a binary numeral to an equivalent decimal numeral. For example, binary 10001011.01 is converted to decimal 139.25. [610.1]

binary-to-hexadecimal conversion. The process of converting a binary numeral to an equivalent hexadecimal numeral. For example, binary 10001011.01 is converted to hexadecimal 8B.4. [610.1]

binary-to-octal conversion. The process of converting a binary numeral to an equivalent octal numeral. For example, binary 10001011.01 is converted to octal 213.2. [610.1]

binary tree. A tree in which each nonterminal node has at most two subtrees. *Note*: B tree is sometimes used incorrectly in reference to a binary tree. This usage is considered deprecated. *See also*: **binary search tree; complete binary tree; full binary tree; *n*-ary tree; weight-balanced tree.** [610.5]

binary tree search. A search in which the items in the set to be searched are placed in a binary tree, and the tree is traversed making key comparisons until the argument is found, or the end of the tree is encountered. *See also*: **binary search; digital tree search.** [610.5]

binary variable. A variable that can assume either of two values or logic states: binary zero (false) or binary one (true). *Syn:* **binary-state variable; Boolean variable; two-state variable; two-valued variable.** [610.1]

binary zero. The "false" binary state, usually represented as 0 or F. *Contrast with:* **binary one.** [610.1]

bind. To assign a value to an identifier. For example, to assign a value to a parameter or to assign an absolute address to a symbolic address in a computer program. *See also:* **dynamic binding; static binding.** [610.12]

bionics. A branch of technology relating the functions, characteristics, and phenomena of living systems to the development of mechanical systems. [610.2]

biquinary. Pertaining to a two-part representation of decimal digits consisting of a binary portion with values 0 or 5, and a quinary portion with values 0 through 4. For example, the decimal digit 7 is coded as 12, which implies 5 + 2. [610.1]

biquinary code. A two-part representation of decimal digits consisting of a binary portion with values 0 or 5 and a quinary portion with values 0 through 4. For example, decimal digit 7 is coded as 12. [610.1]

biquinary coded decimal. Pertaining to a number representation system in which each decimal digit is represented by a biquinary code. [610.1]

biquinary notation. Any notation that uses the biquinary code to represent numbers. [610.1]

biquinary numeration system. A numeration system that alternately uses 2 and 5 as bases. *Note:* The abacus uses a biquinary system. [610.1]

BISAM. Acronym for **basic indexed sequential access method.** [610.5]

bisection. *See*: **binary search.** [610.5]

bit. Acronym for **binary digit.** [610.1, 610.5, 610.12]

bitonic merge. *See*: **bitonic sort.** [610.5]

bitonic sort. A variation on Batcher's parallel sort in which one of the two ordered subsets begins in reverse order and the items to be compared and exchanged are selected from the same subset. *Syn*: **bitonic merge.** [610.5]

bit steering. A microprogramming technique

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052770 1

in which the meaning of a field in a microinstruction is dependent on the value of another field in the microinstruction. *Syn:* **immediate control.** *Contrast with:* **residual control.** *See also:* **two-level encoding.**[610.12]

bit string. A sequence of binary digits; for example, the bit string 0101001. *See also:* **character string.** [610.5]

black box. (1) A system or component whose inputs, outputs, and general function are known but whose contents or implementation are unknown or irrelevant. *Contrast with:* **glass box.** [610.12]
(2) Pertaining to an approach that treats a system or component as in (1). *See also:* **encapsulation.** [610.12]

black box model. A model whose inputs, outputs, and functional performance are known, but whose internal implementation is unknown or irrelevant; for example, a model of a computerized change-return mechanism in a vending machine, in the form of a table that indicates the amount of change to be returned for each amount deposited. *Syn:* **behavioral model; input/output model.** *Contrast with:* **glass box model.** [610.3]

black-box testing. *See:* **functional testing (1).** [610.12]

blank character. A graphic representation of the space character. [610.5]

blip. *See:* **document mark.** [610.2]

block. (1) A group of contiguous storage locations, computer program statements, records, words, characters, or bits that are treated as a unit. *Syn:* **data block.** *See also:* **block-structured language; delimiter.** [610.5, 610.12]
(2) To form a group as in (1). *Contrast with:* **deblock.** [610.5, 610.12]
(3) In text editing and text formatting, one or more contiguous characters or lines of text. *See also:* **block operation.** [610.2]

block allocation. *See:* **paging (1).** [610.12]

block copy. In text editing, an operation that copies a block of text from one point to another within a file or between files, leaving the original block of text intact. *Contrast with:* **block move.** [610.2]

block delete. In text editing, an operation that removes a block of text from a file. [610.2]

block diagram. A diagram of a system, computer, or device in which the principal parts are represented by suitably annotated geometrical figures to show both the functions of the parts and their functional relationships. *Syn:* **configuration diagram; system resources chart.** *See also:* **box diagram; bubble chart; flowchart; graph; input-process-output chart; structure chart.** [610.12]

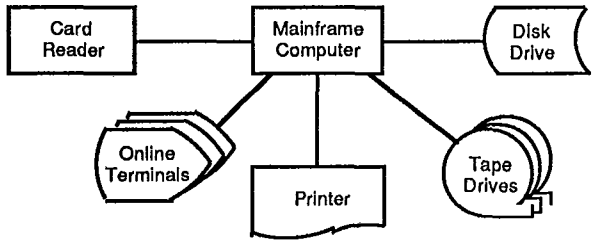


Fig 5
Block Diagram

blocked record. A record that is contained in a block that contains at least one other record. *See also:* **spanned record; unblocked record.** [610.5]

block gap. **See:* **interblock gap.** [610.5]
* Deprecated.

blocking. The process of creating a block from one or more records. [610.5, 610.12]

blocking factor. The number of records, words, characters, or bits in a block. [610.5a, 610.12a]

block length. The number of units in a block. *Syn:* **block size.** [610.5]

block move. In text editing, an operation that moves a block of text from one point to another within a file or between files, deleting the block of text from its original location. *Syn:* **block movement.** *Contrast with:* **block copy.** [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

block movement. *See:* **block move.** [610.2]

block operation. In text editing, an operation that affects a block of text. For example, block copy, block delete, block move. [610.2]

block overhead. Any information, besides the actual data, that is stored with a block; for example, the size and location of the records within the block is considered overhead. *See also*: **loading factor.** [610.5]

block parity. A parity check system capable of detecting and correcting a single error in a binary message. *See also:* **parity check.** [610.1]

block size. *See*: **block length.** [610.5]

block-structured language. A design or programming language in which sequences of statements, called blocks, are defined, usually with begin and end delimiters, and variables or labels defined in one block are not recognized outside that block. Examples include **Ada, ALGOL, PL/I.** *See also:* **structured programming language.** [610.12]

blowback. In micrographics, an enlargement. [610.2]

blurred. Pertaining to elements in an image that are indistinct or not readily discernable. *Contrast with:* **sharp.** [610.4]

boilerplate text. In word processing, standardized previously-stored textual material that may be used to create a new document. *Syn:* **stored paragraph.** [610.2]

bonding. A technique used in database design, in which two or more data items are defined and physically stored together; for example, one might bond data items FIRST-NAME and LAST-NAME. [610.5]

Boolean. Pertaining to the rules of logic formulated by the Irish mathematician George Boole in 1847. [610.1]

Boolean add. *See:* **OR.** [610.1]

Boolean algebra. The binary system of algebra formulated by George Boole, dealing with binary variables and employing the basic logical operators AND, OR, NOT, etc. *Syn:* **Boolean logic; Boolean math.** [610.1]

Boolean calculus. An extension of Boolean algebra that includes time-dependent operators such as BEFORE, DURING, AFTER. [610.1]

Boolean complementation. *See:* **NOT.** [610.1]

Boolean connective. *See:* **Boolean operator.** [610.1]

Boolean function. A switching function in which the number of possible values of the function and each of its independent variables is two. [610.1]

Boolean logic. *See:* **Boolean algebra.** [610.1]

Boolean math. *See:* **Boolean algebra.** [610.1]

Boolean multiplication. *See:* **AND.** [610.1]

Boolean operation. Any operation in which each of the operands and the result take one of two values. [610.1]

Boolean operation table. *See:* **truth table.** [610.1]

Boolean operator. An operator whose operands and results are binary variables. *Syn:* **Boolean connective.** [610.1]

Boolean value. The value of a binary variable; either binary zero or binary one. [610.1]

Boolean variable. *See:* **binary variable.** [610.1]

boot. To initialize a computer system by clearing memory and reloading the operating system. Derived from **bootstrap.** [610.12]

bootstrap. (1) A short computer program that is permanently resident or easily loaded into a computer and whose execution brings a larger program, such as an operating system or its loader, into memory. [610.12] (2) To use a program as in (1). *Syn:* **initial program load.** [610.12]

bootstrap loader. A short computer program used to load a bootstrap. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

border. The set of pixels in a region of a digital image that are adjacent to pixels in the region's complement. *Contrast with:* **interior.** *Syn:* **boundary.** *See also:* **edge; perimeter.** [610.4]

border delineation. *See:* **border detection.** [610.4]

border detection. Any image segmentation technique that identifies borders within a digital image. *Syn:* **border delineation.** [610.4]

borrow. (1) A mathematical process used in subtraction, in which, when the difference in a digit place would be arithmetically negative, the subtraction in that digit place is preceded by increasing the digit in the minuend by the value of the radix, and decreasing the digit in the next higher digit place by one. [610.1]
(2) The value added to the digit place in (1). [610.1]
(3) To perform the process defined in (1). [610.1]

bottom. (1) In a queue, the position of the item that has been in the queue for the shortest time. [610.5]
(2) In a stack, the position of the item that has been in the stack for the longest time. *Contrast with*: **top.** [610.5]

bottom-up. Pertaining to an activity that starts with the lowest-level components of a hierarchy and proceeds through progressively higher levels; for example, bottom-up design; bottom-up testing. *Contrast with:* **top-down.** *See also:* **critical piece first.** [610.12]

boundary. *See:* **border.** [610.4]

boundary alignment. *See*: **alignment.** [610.5]

boundary condition. The values assumed by the variables in a system, model, or simulation when one or more of them is at a limiting value or a value at the edge of the domain of interest. *Contrast with:* **final condition; initial condition.** [610.3]

boundary value. A data value that corresponds to a minimum or maximum input, internal, or output value specified for a system or component. *See also:* **stress testing.** [610.12]

box diagram. A control flow diagram consisting of a rectangle that is subdivided to show sequential steps, if-then-else conditions, repetition, and case conditions. *Syn:* **Chapin chart; Nassi-Shneiderman chart; program structure diagram.** *See also:* **block diagram; bubble chart; flowchart; graph; input-process-output chart; program structure diagram; structure chart.** [610.12]

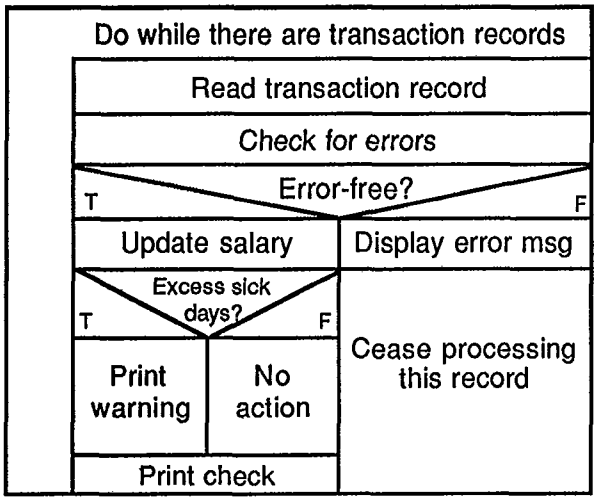


Fig 6
Box Diagram

Boyce/Codd Normal form (BCNF). Developed by R. F. Boyce and E. F. Codd, one of the forms used to characterize relations; a relation is said to be in Boyce/Codd Normal form if every determinant in the relation is or contains a candidate key; that is, no attribute is transitively dependent on any key. *Note*: This is an extension of third normal form. [610.5]

BPAM. Acronym for basic partitioned access method. [610.5]

branch. (1) A computer program construct in which one of two or more alternative sets of program statements is selected for execution. *See also:* **case; jump; go to; if-then-else.** [610.12]
(2) A point in a computer program at which one of two or more alternative sets of program statements is selected for execution. *Syn:* **branchpoint.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

(3) Any of the alternative sets of program statements in (1). [610.12]
(4) To perform the selection in (1). [610.12]
(5) In data management, a synonym for subtree. [610.5]

branch node. *See*: **nonterminal node.** [610.5]

branchpoint. *See:* **branch (2).** [610.12]

branch testing. Testing designed to execute each outcome of each decision point in a computer program. *Contrast with:* **path testing; statement testing.** [610.12]

breakpoint. A point in a computer program at which execution can be suspended to permit manual or automated monitoring of program performance or results. Types include code breakpoint, data breakpoint, dynamic breakpoint, epilog breakpoint, programmable breakpoint, prolog breakpoint, static breakpoint. *Note:* A breakpoint is said to be set when both a point in the program and an event that will cause suspension of execution at that point are defined; it is said to be initiated when program execution is suspended. [610.12]

brightness. In image processing, a value associated with a point of an image, representing the amount of light projected from a scene in a given direction. [610.4]

brother. *See*: **sibling node.** [610.5]

BS. Acronym for the backspace character. [610.5]

BSAM. Acronym for basic sequential access method. [610.5]

B-tree (B tree). (1) A height-balanced search tree of order n in which each node contains keys $\{k_1, k_2, \ldots k_m\}$ in ascending order, where $m <= n - 1$. The i-th subtree of that node contains all the key values falling between k_{i-1} and k_i, with the first subtree containing all key values less than k_1 and the last subtree containing all key values greater than k_m. For example, in the B-tree in Fig 3 below, the lowest notes contain "values less than 10," "11–19," "20–44," "45–59," and "values greater than 60," respectively. *Note*: The height balance of a B-tree is zero. *Syn*: **B-tree index.** *See also*: **B′-tree; B*-tree.** [610.5]

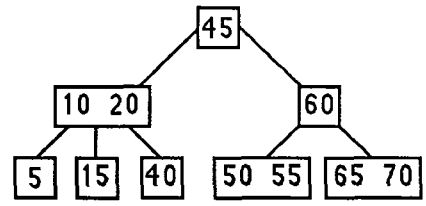


Fig 7
B-Tree of Order 3

(2) A B-tree as in (1) in which every nonterminal node except the root has at least $n/2$ subtrees. *Note*: When a node overflows, it is split into two separate nodes, with the parent node updated accordingly. [610.5]
(3) * *See:* **binary tree.** [610.5]
* Deprecated.

B′-tree. A modified B-tree in which identifiers for all nodes are stored in terminal nodes. [610.5]

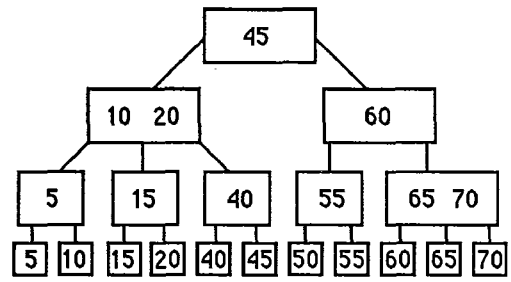


Fig 8
B′-Tree of Order 3

B*-tree. A B-tree in which the root node has between 2 and 2+1 descendants, and each remaining node has between $(2m-1)/3$ and m descendants. That is, two-thirds of the available space in each node is used. *Note*: When a node overflows, keys from that node are moved into one of its sibling nodes if possible; otherwise the node, together with one of its sibling nodes, is split into three nodes. [610.5]

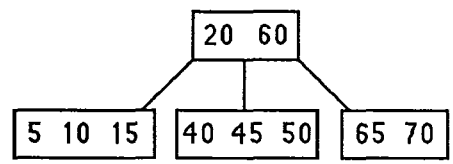


Fig 9
B*-Tree of Order 3

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052774 9

B-tree index. *See*: **B-tree.** [610.5]

bubble chart. A data flow, data structure, or other diagram in which entities are depicted with circles (bubbles) and relationships are represented by links drawn between the circles. *See also:* **block diagram; box diagram; flowchart; graph; input-process-output chart; structure chart.** [610.12]

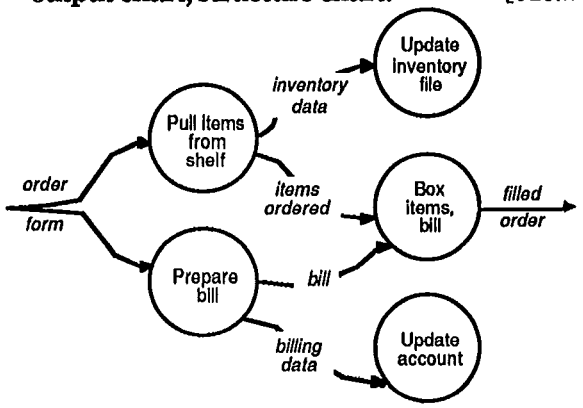


Fig 10
Bubble Chart

bubble sort. An exchange sort in which adjacent pairs of items are compared and exchanged, if necessary, and all passes through the set proceed in the same direction. *Syn*: **exchange selection sort; propagation sort; sifting sort.** *Contrast with*: **cocktail shaker sort.** [610.5]

bucket. (1) An area of storage that may contain more than one record and that is referenced as a whole by some addressing technique. [610.5]
(2) In hashing, a section of a hash table that can hold all records with identical hash values. [610.5]

buffer. (1) A device or storage area used to store data temporarily to compensate for differences in rates of data flow, time of occurrence of events, or amounts of data that can be handled by the devices or processes involved in the transfer or use of the data. *Syn*: **input buffer; input/output area; output buffer.** [610.5, 610.12]
(2) A routine that accomplishes the objectives in (1). [610.12]
(3) To allocate, schedule, or use devices or storage areas as in (1). *See also:* **anticipatory buffering; dynamic buffering; simple buffering.** [610.5, 610.12]

buffered input. Input that is received using buffers. [610.5]

buffer pool. A collection of buffers that can be allocated and used as needed. [610.5]

buffer storage. A storage device that is used as a buffer. *Syn*: **buffer store.** [610.5]

buffer store. *See*: **buffer storage.** [610.5]

bug. *See:* **error; fault.** [610.12]

bug seeding. *See:* **error seeding.** [610.12]

build. An operational version of a system or component that incorporates a specified subset of the capabilities that the final product will provide. [610.12]

built-in check. *See*: **automatic check.** [610.5]

built-in simulation. A special-purpose simulation provided as a component of a simulation language; for example, a simulation of a bank that can be made specific by stating the number of tellers, number of customers, and other parameters. [610.3]

built-in simulator. A simulator that is built-in to the system being modeled; for example, an operator training simulator built into the control panel of a power plant such that the system can operate in simulator mode or in normal operating mode. [610.3]

bulletin board. *See:* **electronic bulletin board.** [610.2]

burn-in period. *See:* **early-failure period.** [610.12]

business data processing (BDP). The use of computers for processing information to support the operational, logistical, and functional activities performed by an organization. [610.2]

business information system. *See:* **management information system.** [610.2]

busy. Pertaining to a system or component that

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

is operational, in service, and in use. *See also:* **down; idle; up.** [610.12]

busy time. In computer performance engineering, the period of time during which a system or component is operational, in service, and in use. *See also:* **down time; idle time; set-up time; up time.** [610.12]

byte. (1) A group of adjacent binary digits operated upon as a unit and usually shorter than a computer word (frequently connotes a group of eight bits). [610.1, 610.5, 610.12]
(2) An element of computer storage that can hold a group of bits as in (1). [610.5, 610.12]
See also: **bit; word.** [610.12]

c. Abbreviation for **centi.** [610.1a]

CAD. Acronym for **computer-aided design.** [610.2]

CADE. Acronym for computer-aided design and engineering. *See:* **computer-aided design; computer-aided engineering.** [610.2a]

CADEM. Acronym for computer-aided design, engineering, and manufacturing. *See:* **computer-aided design; computer-aided engineering; computer-aided manufacturing.** [610.2a]

CADM. Acronym for computer-aided design and manufacturing. *See:* **computer-aided design; computer-aided manufacturing.** [610.2a]

CAE. (1) Acronym for **computer-aided engineering.** [610.2a]
(2) Acronym for **computer-aided education.** *See:* **computer-assisted instruction.** [610.2a]

CAI. (1) Acronym for **computer-assisted instruction.** [610.2a]
(2) Acronym for **computer-aided instruction.** [610.2a]
(3) Acronym for **computer-aided inspection.** [610.2a]

CAL. (1) Acronym for **computer-assisted learning.** *See:* **computer-assisted instruction.** [610.2a]
(2) Acronym for **computer-augmented learning.** *See:* **computer-assisted instruction.** [610.2a]

calc algorithm. *See:* **hash function.** [610.5]

calc chain. *See:* **collision chain.** [610.5]

call. (1) A transfer of control from one software module to another, usually with the implication that control will be returned to the calling module. *Contrast with:* **go to.** [610.12]
(2) A computer instruction that transfers control from one software module to another as in (1) and, often, specifies the parameters to be passed to and from the module. [610.12]
(3) To transfer control from one software module to another as in (1) and, often, to pass parameters to the other module. *Syn:* **cue.**
See also: **call by name; call by reference; call by value; call list; calling sequence.** [610.12]

call by address. *See:* **call by reference.** [610.12]

call by location. *See:* **call by reference.** [610.12]

call by name. A method for passing parameters, in which the calling module provides to the called module a symbolic expression representing the parameter to be passed, and a service routine evaluates the expression and provides the resulting value to the called module. *Note:* Because the expression is evaluated each time its corresponding formal parameter is used in the called module, the value of the parameter may change during the execution of the called module. *Contrast with:* **call by reference; call by value.** [610.12]

call by reference. A method for passing parameters, in which the calling module provides to the called module the address of the parameter to be passed. *Note:* With this method, the called module has the ability to change the value of the parameter stored by the calling module. *Syn:* **call by address; call by location.** *Contrast with:* **call by name; call by value.** [610.12]

call by value. A method of passing parameters, in which the calling module provides to the called module the actual

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052776 2

value of the parameter to be passed. *Note:* With this method, the called module cannot change the value of the parameter as stored by the calling module. *Contrast with:* **call by name; call by reference.** [610.12]

call graph. A diagram that identifies the modules in a system or computer program and shows which modules call one another. *Note:* The result is not necessarily the same as that shown in a structure chart. *Syn:* **call tree; tier chart.** *Contrast with:* **structure chart.** *See also:* **control flow diagram; data flow diagram; data structure diagram; state diagram.** [610.12]

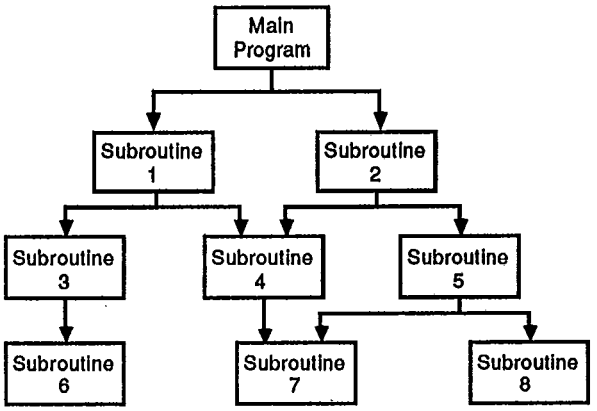


Fig 11
Call Graph

calling sequence. A sequence of computer instructions and, possibly, data necessary to perform a call to another module. [610.12]

call list. The ordered list of arguments used in a call to a software module. [610.12]

call trace. *See:* **subroutine trace.** [610.12]

call tree. *See:* **call graph.** [610.12]

CAM. (1) Acronym for **computer-aided manufacturing.** [610.2]
(2) Acronym for **computer-aided management.** [610.2]

CAN. Abbreviation for the **cancel character.** [610.5]

cancel character (CAN). (1) A control character used by some convention to indicate that the data with which it is associated are in error or are to be disregarded. *Syn:* **ignore character.** [610.5]
(2) An accuracy control character used to indicate that the data with which it is associated are in error or are to be disregarded. [610.5]

candidate key. In a relational data model, any minimal set of attributes within a relation that forms a key that is a determinant of all attributes in the relation. *Note:* In normalization, one of the candidate keys of each relation is chosen as the primary key and the others are known as alternate keys. *See also:* **compound key.** [610.5]

canonical model. A data model that represents the inherent structure of the data, independent of any specific implementations. [610.5]

canonical synthesis. A technique for generating a canonical model from the relations in a database. [610.5]

cardinality. (1) The number of elements in a set. [610.5]
(2) In a relational data model, the number of tuples in a relation. [610.5]

carriage return character (CR). A format effector that causes the print or display position to move to the first position on the same line. *Syn:* **new-line character.** [610.5]

carry. (1) A mathematical process used in addition and subtraction, in which a value is generated when a sum or product in a digit place exceeds the largest number that can be represented in that digit place, and the value is transferred to the next higher digit place for processing there. *See also:* **cascaded carry; complete carry; end-around carry; half carry; high-speed carry; partial carry; standing-on-nines carry.** [610.1]
(2) The value generated in (1). [610.1]
(3) To perform the process defined in (1). [610.1]

cascaded carry. A carry process in which the addition of two numerals results in a partial-sum numeral and a carry numeral that are in turn added together, this process being

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

repeated until no new carries are generated. *Contrast with:* **high-speed carry.** *See also:* **partial carry; partial sum.** [610.1]

Augend	289594
Addend	320607
First Partial Sum	509191
First Partial Carry	101010
Second Partial Sum	600101
Second Partial Carry	010100
True Sum	610201

Fig 12
Example of Cascaded Carry

cascade merge sort. An unbalanced merge sort in which the distribution of the sorted subsets is based on the cascade numbers. *See also:* **polyphase merge sort.** [610.5]

CASE. Acronym for **computer-aided software engineering.** [610.12]

case. A single-entry, single-exit multiple-way branch that defines a control expression, specifies the processing to be performed for each value of the control expression, and returns control in all instances to the statement immediately following the overall construct. *Syn:* **multiple exclusive selective construct.** *Contrast with:* **go to; jump; if-then-else.** *See also:* **multiple inclusive selective construct.** [610.12]

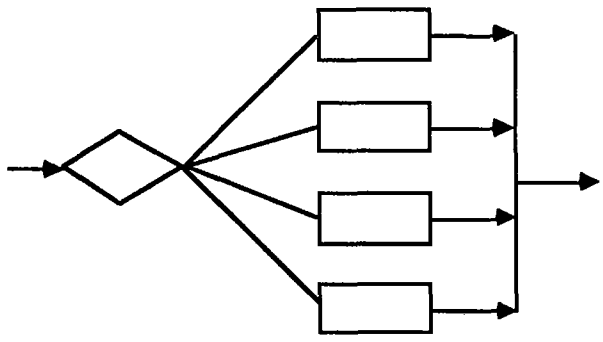


Fig 13
Case Construct

casting out nines. A method of checking addition, subtraction, or multiplication results by dividing decimal values by nine and comparing the remainders. *Syn:* **nines check.** [610.1]

CAT. (1) Acronym for **computer-assisted tomography.** *See:* **computed tomography.** [610.2]
(2) Acronym for **computerized axial tomography.** *See:* **computed tomography.** [610.2]
(3) Acronym for **computer-aided testing.** [610.2]

catalog (catalogue). (1) A directory of the location of files within a system. *See also:* **file directory.** [610.5]
(2) The set of all indices used to reference a file, database, or system. [610.5]
(3) The index to all other indices. *Syn:* **master index.** [610.5]
(4) To enter information about a file, database, or system as in (1) and (2). [610.5]

catastrophic failure. A failure of critical software. [610.12]

category. In pattern recognition, a synonym for **pattern class.** [610.4]

catenate. *See:* **concatenate.** [610.5]

CBCT. Acronym for **customer-bank communication terminal.** *See:* **automated teller machine.** [610.2]

CBE. Acronym for **computer-based education.** *See:* **computer-based instruction.** [610.2]

CBL. Acronym for **computer-based learning.** *See:* **computer-based instruction.** [610.2]

CCB. (1) Acronym for **configuration control board.** [610.12]
(2) Acronym for **change control board.** *See:* **configuration control board.** [610.12]

CDP. Acronym for **commercial data processing.** [610.2]

CDR. Acronym for **critical design review.** [610.12]

ceiling. The result obtained by rounding a number up to the nearest integer. For example, the ceiling of 5.3 is 6. *Contrast with:* **floor.** [610.1]

cell. *See:* **data element; data item.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

center. In text formatting, to format one or more lines of text so that the left and right margins are equal in size. [610.2]

centi (c). A prefix indicating one hundredth (10^{-2}). [610.1a]

certification. (1) A written guarantee that a system or component complies with its specified requirements and is acceptable for operational use. For example, a written authorization that a computer system is secure and is permitted to operate in a defined environment. [610.12]
(2) A formal demonstration that a system or component complies with its specified requirements and is acceptable for operational use. [610.12]
(3) The process of confirming that a system or component complies with its specified requirements and is acceptable for operational use. [610.12]

chain. (1) A sequence of bits used to construct a binary code. *See also:* **chain code.** [610.1]
(2) *See*: **circularly-linked list.** [610.5]
(3) **See*: **linked list.** [610.5]
* Deprecated.

chain code. An arrangement in a cyclic sequence of some or all of the possible different n-bit words in which adjacent words are related such that each word is derivable from its neighbor by displacing the bits one place to the left or right, dropping the leading bit, and inserting a bit at the end. The value of the inserted bit needs only to meet the requirement that a word must not recur before the cycle is complete. For example, 000 001 010 101 011 111 110 100 000. [610.1]

chained list. *See*: **linked list.** [610.5]

chain field. *See*: **link field.** [610.5]

chaining. A method for storing records in which each record has a link field that is used to access subsequent records. [610.5]

chaining search. A search in which each item contains a means for locating the next item to be considered in the search. [610.5]

change control board. *See:* **configuration control board.** [610.12]

change control. *See:* **configuration control.** [610.12]

change detection. An image processing technique in which the pixels of two registered images are compared to detect differences. [610.4]

change dump. A selective dump of those storage locations whose contents have changed since some specified time or event. *Syn:* **differential dump.** *See also:* **dynamic dump; memory dump; postmortem dump; selective dump; snapshot dump; static dump.** [610.12]

change transaction. A transaction that causes information in a master file to be changed. *See also:* **add transaction; delete transaction; null transaction; update transaction.** [610.2]

channel capacity. The maximum amount of information that can be transferred on a given channel per unit of time; usually measured in bits per second or in baud. *See also:* **memory capacity; storage capacity.** [610.12]

Chapin chart. *See:* **box diagram.** [610.12]

character. A letter, digit, or other symbol that is used to represent information. *See also*: **alphabetic; alphanumeric.** [610.1, 610.5, 610.12]

character boundary. In character recognition, the largest rectangle, with a side parallel to the document reference edge, whose sides are tangential to a given character outline. [610.2]

character code. A code that uses unique numeric values to represent character data; for example, in ASCII the hexadecimal value 40 is used to represent the character "@." [610.5]

character-deletion character. A character within a line of terminal input specifying that it and the immediately preceding character are to be removed from the line; for

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

example, if "\" is the character-deletion character in the string "ABD\C," the following would appear on the terminal: "ABC." *See also*: **line-deletion character.** [610.5]

character fill. To insert into a storage medium, as often as necessary, the representation of a specified filler character that does not itself convey data but that may delete unwanted data or initialize storage. *See also*: **zero fill.** [610.5]

character font. A set of graphic characters that are of the same size and style. *See also:* **font disk; hand-printed character font; OCR-A; OCR-B; optical font.** *Syn:* **type font.** [610.2]

character outline. In character recognition, the graphic pattern established by the stroke edges of a character. [610.2]

character recognition. The use of pattern recognition techniques to identify characters by automatic means. *See also:* **magnetic ink character recognition; omni-font character recognition; optical character recognition; single-font character recognition.** [610.2]

character representation system. *See:* **character set.** [610.1]

character set. (1) The set of all characters that is defined for a given system. [610.1a, 610.5] (2) A finite set of unique characters upon which agreement has been reached and that is considered complete for some purpose; for example, all the letters, numbers, and symbols used in a language. [610.5] (3) A finite set of unique representations called characters, made to denote and distinguish data; for example, the 26 letters of the English alphabet; 0 and 1 of the boolean alphabet; the set of signals in the Morse code alphabet; and the 128 ASCII characters. *See also*: **alphabetic character set; alphanumeric character set; coded character set; numeric character set.** [610.5]

character spacing reference line. In character recognition, a vertical line that is used to evaluate the horizontal spacing of characters. *Note:* It may be a line that equally divides the distance between the sides of a character boundary or that coincides with the centerline of a vertical stroke. [610.2]

character string. A sequence of characters; for example, the character string 72ZABC. *See also*: **bit string.** [610.5]

character string picture data. Picture data that is associated with a picture specification that specifies at least one alphanumeric character. [610.5]

character stroke. In optical character recognition, a line, point, arc, or other mark used as a portion of a graphic character. For example, the dot over the letter i or the cross of the letter t. [610.2]

character type. A data type whose members can assume the values of specified characters and can be operated on by character operators, such as concatenation. *Contrast with:* **enumeration type; integer type; logical type; real type.** [610.12]

character variable. A variable that may assume values of any character within some character set. [610.5]

characteristic overflow. *See:* **exponent overflow.** [610.1]

characteristic underflow. *See:* **exponent underflow.** [610.1]

characteristic. (1) The integer part of a logarithm. *Contrast with:* **mantissa (1).** [610.1] (2) For floating point arithmetic, *see:* **exponent (2).** [610.1] (3) *See:* **data characteristic; software characteristic.** [610.12]

check. To verify the accuracy of data transmitted, manipulated, or stored by any unit or device in a computer. *See also*: **automatic check; check character; check key; consistency check; echo check; sequence check.** [610.5]

check bit. A binary check digit. For example, a parity bit. [610.1]

check character. (1) A character used for the

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

purpose of performing a check. [610.5]
(2) A single character from a check key. [610.5]

check digit. One of a set of redundant digits in a word, byte, character, or message that depends upon the remaining digits in such a fashion that if a digit changes, the error can be detected. [610.1]

check key. A key that is used for the purpose of performing a check; for example, in the following example the check key is equal to the sum of the first and last digit in field x; this check key could be used to ensure that field x is accurate and complete. [610.5]

record number	field x	check key
1	0125	5
2	1136	7
3	2228	10

checkout. Testing conducted in the operational or support environment to ensure that a software product performs as required after installation. [610.12]

checkpoint. A point in a computer program at which program state, status, or results are checked or recorded. [610.12]

check problem. A test or problem that is chosen to determine whether an operations or computer program is operating properly. [610.5]

check sum. A sum obtained by adding the digits in a numeral, or group of numerals, usually without regard to meaning, position, or significance. This sum may be compared with a previously computed value to verify that no errors have occurred. *See also:* **sideways sum; summation check.** [610.1a]

chief programmer team. A software development group that consists of a chief programmer, a backup programmer, a secretary/librarian, and additional programmers and specialists as needed, and that employs procedures designed to enhance group communication and to make optimum use of each member's skills. *See also:* **backup programmer; chief programmer; egoless programming.** [610.12]

chief programmer. The leader of a chief programmer team; a senior-level programmer whose responsibilities include producing key portions of the software assigned to the team, coordinating the activities of the team, reviewing the work of the other team members, and having an overall technical understanding of the software being developed. *See also:* **backup programmer; chief programmer team.** [610.12]

child node. In a tree, a descendant node having a given node as its parent node. *Syn:* **daughter; son.** *Contrast with:* **parent node.** *See also:* **sibling node.** [610.5]

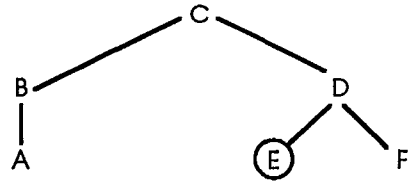


Fig 14
E is a Child Node of Node D

child segment. In a hierarchical database, a segment that has a parent segment and that is dependent on that segment for its existence. *Note:* If the parent segment is deleted, the child segment must be deleted. *Contrast with:* **parent segment.** *See also:* **dependent segment; logical child segment; physical child segment; twin segment.** [610.5]

Chinese binary. *See:* **column binary.** [610.1]

CI. Acronym for **configuration item.** [610.12]

CIM. (1) Acronym for **computer input microfilm.** [610.2]
(2) Acronym for **computer-integrated manufacturing.** [610.2]

cine-oriented image. In micrographics, an image appearing on a roll of microfilm in such a manner that the top edge of the image is perpendicular to the long edge of the film. *Contrast with:* **comic-strip oriented image.** *Syn:* **motion-picture display; portrait image.** [610.2]

circular list. *See:* **circularly-linked list.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

circularly linked list. A linked list in which the last item contains a pointer to the first item. *Syn*: **chain (2); circular list; ring.** [610.5a]

circular shift. A variation of a logical shift in which the digits moved out of one end of a register, word, or numeral are returned at the other end. For example, + 231.702 shifted two places to the left becomes 3170.2+2. *Note:* A circular shift may be applied to the multiple precision representation of a number. *Syn:* **cyclic shift; end-around shift; end-around carry shift; ring shift; rotate.** [610.1]

class. *See:* **pattern class.** [610.4]

classifier. *See:* **decision rule.** [610.4]

clear. To set a variable, register, or other storage location to zero, blank, or other null value. *See also:* **initialize; reset.** [610.12]

clear area. In character recognition, a specified area that is to be kept free of printing or other markings that are not related to machine readings. [610.2]

closed curve. A curve whose beginning and ending points are the same point. [610.4]

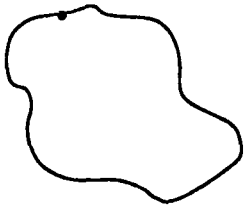


Fig 15
Illustration of Closed Curve

closed loop. A loop that has no exit and whose execution can be interrupted only by intervention from outside the computer program or procedure in which the loop is located. *Contrast with:* **UNTIL; WHILE.** [610.12]

closed-loop control. Pertaining to a control system in which the output is measured and compared with a standard representing the acceptable range, and any deviation from the standard is fed back into the system in a way that will reduce the deviation. *Syn:* **feedback control.** *Contrast with:* **open-loop control.** [610.2]

closed subroutine. A subroutine that is stored at one given location rather than being copied into a computer program at each place that it is called. *Contrast with:* **open subroutine.** [610.12]

cluster. (1) In image processing, a set of pixels in a digital image that are close to one another and similar in some way. [610.4] (2) In pattern recognition, a set of points in a feature space that are similar in some way. [610.4]

cluster analysis. (1)The detection and description of clusters in a digital image. [610.4] (2) The detection and description of clusters in a feature space. [610.4]

clustered word processing. Word processing performed on a system composed of multiple work stations, each with its own memory but operating under the control of a master work station. *Contrast with:* **dedicated word processing; shared-logic word processing; shared-resource word processing; stand-alone word processing.** [610.2]

CM. Acronym for **configuration management.** [610.12]

CMC. Acronym for **code for magnetic characters.** *See:* **magnetic ink character recognition.** [610.2]

CMI. Acronym for **computer-managed instruction.** [610.2]

CNC. Acronym for **computer numerical control.** [610.2]

coalesce. To combine two or more sets into one set. *See also*: **collate (2); merge.** [610.5a]

cocktail shaker sort. An exchange sort in which adjacent pairs of items are compared and exchanged, if necessary, and alternate passes through the set proceed in opposite directions. *Contrast with*: **bubble sort.** [610.5]

CODASYL. Acronym for **Conference on Data Systems Languages.** An organization that

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

establishes standards for database structures. [610.5]

CODASYL database. A database that adheres to the standards established by the Database Task Group of CODASYL. *Note*: A network database is generally accepted to be synonymous with a CODASYL database. [610.5]

CODASYL model. A network database model defined by the CODASYL organization. The CODASYL model is based on sets that are used to specify associations between different record types that exist in a database. *Syn*: **flex model.** [610.5]

CODASYL set. *See:* **set.** [610.5]

code. (1) In software engineering, computer instructions and data definitions expressed in a programming language or in a form output by an assembler, compiler, or other translator. *See also:* **source code; object code; machine code; microcode.** [610.12]
(2) To express a computer program in a programming language. [610.12]
(3) A set of rules used to convert data from one form of representation to another. *Syn*: **coding scheme; data code; data element tag.** [610.1, 610.5]
(4) Data that have been converted from one form of representation to another, using a set of rules as in (1). *See also*: **code set; coded representation; symbol.** *Syn*: **encoded data.** [610.5]
(5) Data that have been expressed in symbolic form. [610.1, 610.5]
(6) A character or bit pattern that is assigned a particular meaning; for example, a status code. [610.5, 610.12]
(7) To convert data from one form of representation to another, using a set of rules as in (1). *See also*: **decode; encode.** [610.5]
(8) To represent data in symbolic form. [610.1, 610.5]
(9) * *Syn*: **code set.** [610.5]
* Deprecated.

code breakpoint. A breakpoint that is initiated upon execution of a given computer instruction. *Syn:* **control breakpoint.** *Contrast with:* **data breakpoint.** *See also:* **dynamic breakpoint; epilog breakpoint; programmable breakpoint; prolog breakpoint; static breakpoint.** [610.12]

code converter. A device or system that changes the representation of data from one code to another. [610.5]

coded arithmetic data. Data stored in a form that is acceptable for arithmetic calculations without conversion to an intermediate form; for example, data stored in integer form. [610.5]

coded character set. A set of characters for which coded representations exist. *Syn*: **coded representation; code set.** [610.5]

coded decimal. *See:* **binary-coded decimal.** [610.1]

code-decode table. A table that identifies a correspondence between encoded and decoded data items. *Syn*: **encode-decode table.** [610.5a]

code distance. *See:* **Hamming distance.** [610.1]

coded representation. The result of applying a code to a particular item of data. For example, the designation ORY for Paris International Airport, obtained by applying the international three-letter code for airports. *Syn*: **code value.** *See*: **coded character set.** [610.5]

code extension character. Any control character used to indicate that one or more of the succeeding coded representations are to be interpreted according to a different code or according to a different coded character set. [610.5]

code for magnetic characters (CMC). A set of rules used in magnetic ink character recognition. [610.2]

code generator. (1) A routine, often part of a compiler, that transforms a computer program from some intermediate level of representation (often the output of a root compiler or parser) into a form that is closer to the language of the machine on which the program will execute. [610.12]
(2) A software tool that accepts as input the requirements or design for a computer pro-

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

gram and produces source code that implements the requirements or design. *Syn:* **source code generator.** *See also:* **application generator.** [610.12]

code inspection. *See:* **inspection.** [610.12]

code of ethics standard. A standard that describes the characteristics of a set of moral principles dealing with accepted standards of conduct by, within, and among professionals. [610.12]

code review. A meeting at which software code is presented to project personnel, managers, users, customers, or other interested parties for comment or approval. *Contrast with:* **design review; formal qualification review; requirements review; test readiness review.** [610.12]

code set. The complete set of coded representations used by a particular code. For example, the set of three-letter codes used to represent airports. [610.1, 610.5]

code trace. *See:* **execution trace.** [610.12]

code value. *See*: **coded representation.** [610.5]

coding scheme. *See*: **code (3).** [610.5]

coding. (1) In software engineering, the process of expressing a computer program in a programming language. [610.12]
(2) The transforming of logic and data from design specifications (design descriptions) into a programming language. [610.12]
See also: **software development process.** [610.12]

coenetic variable. In modeling, a variable that affects both the system under consideration and that system's environment. [610.3]

cohesion. The manner and degree to which the tasks performed by a single software module are related to one another. Types include coincidental, communicational, functional, logical, procedural, sequential, and temporal. *Syn:* **module strength.** *Contrast with:* **coupling.** [610.12]

coincidental cohesion. A type of cohesion in which the tasks performed by a software module have no functional relationship to one another. *Contrast with:* **communicational cohesion; functional cohesion; logical cohesion; procedural cohesion; sequential cohesion; temporal cohesion.** [610.12]

collate. (1) to compare and merge two or more similarly ordered or sequenced sets onto one ordered set. For example, to arrange the set 1, 4, 9, 12, 18 and the set 2, 5, 10, 19 as the single set 1, 2, 4, 5, 9, 10, 12, 18, 19. [610.1]
(2) To arrange items from two or more ordered subsets into one or more other subsets. The resulting subsets will commonly contain at least one item from each of the original subsets and may be ordered in some specified order that is not necessarily the order of any of the original subsets. *See also:* **coalesce; collating sequence (1); merge.** *Contrast with*: **decollate.** [610.5]

collating sequence. (1) A sequence assigned to a set of items such that any two sets that are in that assigned order can be collated. [610.5]
(2) A specified arrangement of the items in a set used in sequencing. *Syn*: **sequence (5).** [610.5]
(3) * *See*: **order (2).** [610.5]
* Deprecated.

collating significance. Any attribute of a set that may be used to define a specified arrangement to be used in collating. [610.5]

collation. *See:* **AND.** [610.1]

collision. In hashing, the occurrence of the same hash value for two or more different keys. *Syn*: **hash clash.** *See also*: **synonym.** [610.5]

collision chain. A list used in hashing to hold all the keys for which the hash address is identical. *Syn*: **calc chain.** [610.5]

collision resolution. In hashing, the process of applying further calculations or other means to resolve a collision. Methods include open-address hashing, separate chaining, and the use of buckets. *Syn*: **rehashing.** [610.5]

column. A vertically corresponding set of

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

entries in a table. *See also*: **attribute (2).** *Contrast with:* **row.** [610.5]

column binary. Pertaining to the binary representation of data in which adjacent positions in a column correspond to adjacent binary digits. For example, each column in a 12-row card may be used to represent 12 consecutive bits of a binary word. *Syn:* **Chinese binary.** *Contrast with:* **row binary.** [610.1]

column-major order. A method for storing the elements of a matrix in computer memory, in which elements are ordered in a column-by-column manner, that is, all elements of column 1, followed by all elements of column 2, etc. *Contrast with*: **row-major order.** [610.5]

column sort. *See*: **distribution sort.** [610.5]

column split. The capability of a punch card device to read or punch two parts of a card column independently. [610.1]

column vector. A matrix with only one column. That is, a matrix of size m-by-1. *Contrast with*: **row vector.** [610.5]

COM device. *See:* **computer output microfilmer.** [610.2]

COM. Acronym for **computer output microfilm.** [610.2]

combination. An unordered sequence of items chosen from a set. *See also*: **forbidden combination.** *Contrast with*: **permutation.** [610.5]

comic-strip oriented image. In micrographics, an image appearing on a roll of microfilm in such a manner that the top edge of the image is parallel to the long edge of the film. *Contrast with:* **cine-oriented image.** *Syn:* **landscape image.** [610.2]

command. An expression that can be input to a computer system to initiate an action or affect the execution of a computer program; for example, the "log on" command to initiate a computer session. [610.12]

command character. *See*: **control character.** [610.5]

command language. A language used to express commands to a computer system. *See also:* **command-driven.** [610.12]

command-driven. Pertaining to a system or mode of operation in which the user directs the system through commands. *Contrast with:* **menu-driven.** [610.12]

comment. Information embedded within a computer program, job control statements, or a set of data, that provides clarification to human readers but does not affect machine interpretation. [610.12]

commercial character. (1) One of the set of characters used commonly in commercial operations; for example, CR (credit) and DB (debit). [610.5]
(2) A character within a picture specification that represents one of the characters as in (1). [610.5]

commercial data processing. Data processing performed to support a commercial organization or function. [610.2]

common. *See:* **common storage.** [610.12]

common area. *See:* **common storage.** [610.12]

common block. *See:* **common storage.** [610.12]

common coupling. *See:* **common-environment coupling.** [610.12]

common data. *See:* **global data.** [610.12]

common-environment coupling. A type of coupling in which two software modules access a common data area. *Syn:* **common coupling.** *Contrast with:* **content coupling; control coupling; data coupling; hybrid coupling; pathological coupling.** [610.12]

common storage. A portion of main storage that can be accessed by two or more modules in a software system. *Syn:* **common area; common block.** *See also:* **global data.** [610.12]

communicational cohesion. A type of cohesion in which the tasks performed by a software module use the same input data or contribute

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

to producing the same output data. *Contrast with:* **coincidental cohesion; functional cohesion; logical cohesion; procedural cohesion; sequential cohesion; temporal cohesion.** [610.12]

communication control character. *See:* **transmission control character.** [610.5]

compaction. In microprogramming, the process of converting a microprogram into a functionally equivalent microprogram that is faster or shorter than the original. *See also:* **local compaction; global compaction.** [610.12]

comparator. A software tool that compares two computer programs, files, or sets of data to identify commonalities or differences. Typical objects of comparison are similar versions of source code, object code, data base files, or test results. [610.12]

compare. (1) To examine a quantity for the purpose of determining its relationship to zero. [610.1a]
(2) To examine two items to determine their relative magnitudes, their relative positions in a given sequence, or whether they are identical. [610.5]

comparison. (1) The process of examining two or more items for identity, similarity, equality, relative magnitude, or for order in a sequence. [610.5]
(2) The result of such an examination as in (1). [610.5]

compatibility. (1) The ability of two or more systems or components to perform their required functions while sharing the same hardware or software environment. [610.12]
(2) The ability of two or more systems or components to exchange information. [610.12]
See also: **interoperability.** [610.12]

compile. To translate a computer program expressed in a high-order language into its machine language equivalent. *Contrast with:* **assemble; decompile; interpret.** [610.12]

compile-and-go. An operating technique in which there are no stops between the compiling, linking, loading, and execution of a computer program. [610.12]

compiler. A computer program that translates programs expressed in a high-order language into their machine language equivalents. *Contrast with:* **assembler; interpreter.** *See also:* **cross-compiler; incremental compiler; root compiler.** [610.12]

compiler code. Computer instructions and data definitions expressed in a form that can be recognized and processed by a compiler. *Contrast with:* **assembly code; interpretive code; machine code.** [610.12]

compiler compiler. *See:* **compiler generator.** [610.12]

compiler generator. A translator or interpreter used to construct part or all of a compiler. *Syn:* **compiler compiler; metacompiler.** [610.12]

complement. (1) A numeral derived from a given numeral by a specified subtraction rule. Often used to represent the negative of the number represented by the given numeral. *See also:* **radix complement; diminished-radix complement.** [610.1]
(2) In image processing, all points in an image that do not belong to a given subset of the image. [610.4]

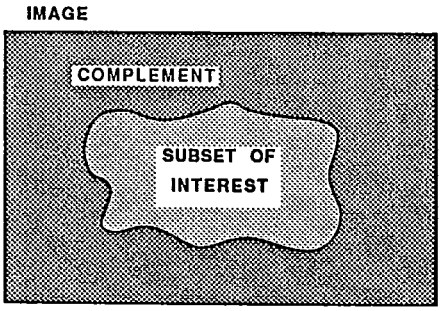


Fig 16
Illustration of Complement

complementary operation. Two Boolean operations are complementary if the result of one operation is the negation of the result of the other, for all combinations of operands. For example, the AND and NAND operations are complementary. *Contrast with:* **dual operation.** [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052786 5

complementary operator. *See:* **NOT.** [610.1]

complementation. The process of obtaining a complement. [610.1]

complement base. The numeral from which a given numeral is subtracted to obtain its complement. [610.1]

complemented representation. A positional notation system in which negative numbers are represented by their complements and positive numbers are represented in their usual form. *See also:* **twos-complement notation.** [610.1]

complement on *n*. *See:* **radix complement.** [610.1]

complement on *n*–1. *See:* **diminished-radix complement.** [610.1]

complement on nine. *See:* **nines complement.** [610.1]

complement on one. *See:* **ones complement.** [610.1]

complement on ten. *See:* **tens complement.** [610.1]

complement on two. *See:* **twos complement.** [610.1]

complete binary tree. A complete tree of order 2. *Note*: The nodes in the tree can be read sequentially from left to right; top to bottom. *Syn*: **full binary tree.** [610.5]

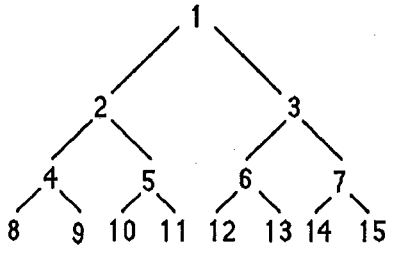


Fig 17
Complete Binary Tree

complete carry. A carry process in which the carry digits are transferred and processed as they occur. *Contrast with:* **partial carry.** [610.1]

complete tree. A tree of order n in which each node has exactly n subtrees. *Syn*: **full tree.** *See also*: **heap.** [610.5]

completion code. A code communicated to a job stream processor by a batch program to influence the execution of succeeding steps in the input stream. [610.12]

complex data. Numeric data used to represent complex numbers. [610.5]

complex data structure. *See*: **nonprimitive data structure.** [610.5]

complexity. (1) The degree to which a system or component has a design or implementation that is difficult to understand and verify. *Contrast with:* **simplicity.** [610.12] (2) Pertaining to any of a set of structure-based metrics that measure the attribute in (1). [610.12]

complex number. A number consisting of a real part (a) and an imaginary part (b), expressed in the form $a + bi$, where $i^2 = -1$. [610.1, 610.5]

component. One of the parts that make up a system. A component may be hardware or software and may be subdivided into other components. *Note:* The terms "module," "component," and "unit" are often used interchangeably or defined to be subelements of one another in different ways depending upon the context. The relationship of these terms is not yet standardized. [610.12]

component data element. A component of a data structure. *Syn*: **component element; element.** [610.5]

component element. *See*: **component data element.** [610.5]

component standard. A standard that describes the characteristics of data or program components. [610.12]

component testing. Testing of individual hardware or software components or groups of related components. *Syn:* **module testing.** *See also:* **integration testing; interface testing; system testing; unit testing.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

composite data element. A data element that contains two or more data elements that can be referred to either collectively or individually; for example, a data element named "date of birth" containing data elements "year," "month," and "day." *See also*: **data aggregate.** *Syn*: **data chain; molecular data element.** *Contrast with*: **atomic data element.** [610.5]

composite type. A data type each of whose members is composed of multiple data items. For example, a data type called PAIRS whose members are ordered pairs (x,y). *Contrast with:* **atomic type.** [610.12]

compound key. A candidate key consisting of more than one attribute. [610.5]

compound list. *See*: **list structure.** [610.5]

compression. *See:* **image compression.** [610.4]

computational data. *See*: **fixed-point data.** [610.5]

computational model. A model consisting of well-defined procedures that can be executed on a computer; for example, a model of the stock market, in the form of a set of equations and logic rules. [610.3]

computed tomography (CT). A medical diagnostic technique in which a computer is used to produce an image of cross-sections of the human body by using measured attenuation of X rays through a cross-section of the body. *Syn:* **computer-assisted tomography.** [610.2]

computer-aided design (CAD). The use of computers to aid in design layout and analysis. May include modeling, analysis, simulation, or optimization of designs for production. Often used in combinations such as CAD/CAM. *See also:* **computer-aided engineering; computer-aided manufacturing; design automation.** [610.2]

computer-aided education (CAE). *See:* **computer-assisted instruction.** [610.2]

computer-aided engineering (CAE). The use of computers to aid in engineering analysis and design. May include solution of mathematical problems, process control, numerical control, and execution of programs performing complex or repetitive calculations. *See also:* **computer-aided design; computer-aided manufacturing.** [610.2]

computer-aided inspection (CAI). The use of computers to inspect manufactured parts. *Syn:* **mechanical inspection.** [610.2]

computer-aided instruction (CAI). *See:* **computer-assisted instruction.** [610.2]

computer-aided management (CAM). The application of computers to business management activities. For example, database management, control reporting, and information retrieval. *See also:* **decision support system; management information system.** [610.2]

computer-aided manufacturing (CAM). The use of computers and numerical control equipment to aid in manufacturing processes. May include robotics, automation of testing, management functions, control, and product assembly. Often used in combinations such as CAD/CAM. *See also:* **computer-aided design; computer-aided engineering.** [610.2]

computer-aided page makeup. The use of computers to automate the formation of text and graphics into discrete camera-ready pages. *See also:* **computer-aided typesetting, photocomposition.** [610.2]

computer-aided software engineering (CASE). The use of computers to aid in the software engineering process. May include the application of software tools to software design, requirements tracing, code production, testing, document generation, and other software engineering activities. [610.12]

computer-aided testing (CAT). The use of computers to test manufactured parts. [610.2]

computer-aided typesetting. The use of computers at any stage of the document composition process. This may involve text formatting, input from a word processing system, or computer-aided page makeup. *Syn:* **computer typesetting.** [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

computer-assisted instruction (CAI). The use of computers to present instructional material and to accept and evaluate student responses. *Syn:* **computer-aided education; computer-aided instruction; computer-assisted learning; computer-augmented learning.** *See also:* **computer-based instruction.** [610.2]

computer-assisted learning (CAL). *See:* **computer-assisted instruction.** [610.2]

computer-assisted tomography (CAT). *See:* **computed tomography.** [610.2]

computer-augmented learning (CAL). *See:* **computer-assisted instruction.** [610.2]

computer-based education (CBE). *See:* **computer-based instruction.** [610.2]

computer-based instruction (CBL). The use of computers to support any process involving human learning. *Syn:* **computer-based education; computer-based learning.** [610.2]

computer-based learning (CBL). *See:* **computer-based instruction.** [610.2]

computer-based simulation. A simulation that is executed on a computer. *Syn:* **machine-centered simulation.** *Contrast with:* **human-centered simulation.** [610.3]

computer conferencing. A form of teleconferencing that allows one or more users to exchange messages on a computer network. *See also:* **video conferencing.** [610.2]

computer input microfilm (CIM). The input to a process that converts data contained on microform into machine-readable data. [610.2]

computer instruction. (1) A statement in a programming language, specifying an operation to be performed by a computer and the addresses or values of the associated operands; for example, Move A to B. *See also:* **instruction format; instruction set.** [610.12] (2) Loosely, any executable statement in a computer program. [610.12]

computer-integrated manufacturing (CIM). Use of an integrated system of computer-controlled manufacturing centers. The centers may use robotics, design automation, or CAD/CAM technologies. *See also:* **flexible manufacturing system.** [610.2]

computerized axial tomography (CAT). *See:* **computed tomography.** [610.2]

computer language. A language designed to enable humans to communicate with computers. *See also:* **design language; query language; programming language.** *Note:* P610.13 will define specific computer languages. [610.12]

computer literacy. An understanding of the capabilities, operation, and applications of computers. [610.2]

computer-managed instruction (CMI). The use of computers for management of student progress. Activities may include record keeping, progress evaluation, and lesson assignment. *See also:* **computer-based instruction.** [610.2]

computer numerical control (CNC). Numerical control in which one or more machines that produce manufactured parts are linked together via a single computer. [610.2]

computer output microfilm (COM). The end result of a process that converts and records data from a computer directly to a microform. [610.2]

computer output microfilmer. A device for producing computer output microfilm. *Syn:* **COM device.** [610.2]

computer performance evaluation. An engineering discipline that measures the performance of computer systems and investigates methods by which that performance can be improved. *See also:* **system profile; throughput; utilization; workload model.** [610.12]

computer program. A combination of computer instructions and data definitions that enable computer hardware to perform computational or control functions. *See also:* **software.** [610.5, 610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

computer program abstract. A brief description of a computer program that provides sufficient information for potential users to determine the appropriateness of the program to their needs and resources. [610.12]

computer program component (CPC).* *See:* **computer software component.** [610.12]
*Deprecated.

computer program configuration item (CPCI).* *See:* **computer software configuration item.** [610.12]
*Deprecated.

computer resource allocation. The assignment of computer resources to current and waiting jobs; for example, the assignment of main memory, input/output devices, and auxiliary storage to jobs executing concurrently in a computer system. *See also:* **dynamic resource allocation; storage allocation.** [610.12]

computer resources. The computer equipment, programs, documentation, services, facilities, supplies, and personnel available for a given purpose. *See also:* **computer resource allocation.** [610.12]

computer security. *Note:* P610.9 will define terminology pertaining to computer security. [610.12, P610.9]

computer simulation. A simulation of the operation of a computer. *See also:* **computer-based simulation.** [610.3]

computer software component (CSC). A functionally or logically distinct part of a computer software configuration item, typically an aggregate of two or more software units. [610.12]

computer software configuration item (CSCI). An aggregation of software that is designated for configuration management and treated as a single entity in the configuration management process. *Contrast with:* **hardware configuration item.** *See also:* **configuration item.** [610.12]

computer system. A system containing one or more computers and associated software. [610.12]

computer typesetting (CTS). *See:* **computer-aided typesetting.** [610.2]

computer word. *See:* **word.** [610.1, 610.5, 610.12]

computing center. A facility designed to provide computer services to a variety of users through the operation of computers and auxiliary hardware and through services provided by the facility's staff. [610.12]

concatenate. To append one item to the end of another so as to form a single unit in a contiguous pattern. For example, if we concatenate 'AP' with 'PLE,' the result is 'APPLE.' *Syn:* **catenate.** [610.5]

concatenated key. (1) A key derived from the concatenation of two or more keys. *Syn:* **fully concatenated key; multifield key.** [610.5]
(2) A concatenation of the keys for the first N segments found in a hierarchical path. For example, in the structure below, the concatenated key for segment x is "AFINANCE006." [610.5]

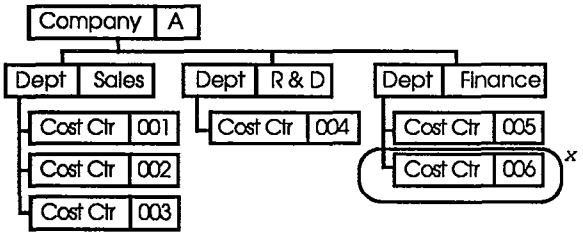


Fig 18
Concatenated Key

concave. Pertaining to a region for which at least one straight line segment between two points of the region is not entirely contained within the region. *Contrast with:* **convex.** [610.4]

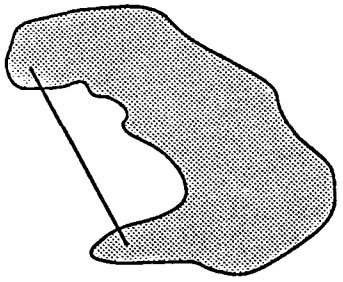


Fig 19
Illustration of Concave

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052790 7

concept phase. (1) The period of time in the software development cycle during which the user needs are described and evaluated through documentation (for example, statement of needs, advance planning report, project initiation memo, feasibility studies, system definition, documentation, regulations, procedures, or policies relevant to the project). [610.12]
(2) The initial phase of a software development project, in which the user needs are described and evaluated through documentation (for example, statement of needs, advance planning report, project initiation memo, feasibility studies, system definition, documentation, regulations, procedures, or policies relevant to the project). [610.12]

conceptual data definition language. A data definition language used to describe the format layout and contents of all data stored in a database, the result of which is a conceptual schema. *Note*: May also include authorization levels and validation procedures. [610.5]

conceptual design. The process of developing a conceptual schema for a database. [610.5]

conceptual model. *See*: **conceptual schema.** [610.5]

conceptual record. A record within a conceptual view. [610.5]

conceptual schema. (1) A description of the format and layout of the entire data contents of a database. *Note*: The schema is written using a conceptual data definition language. It may include authorization levels and validation procedures. [610.5]
(2) The comprehensive, logical description of the information environment in which an enterprise exists, free of both the physical structure and application systems considerations. *Syn*: **conceptual model; enterprise view.** *Contrast with*: **external view; internal schema.** [610.5]

conceptual view. The format and layout of the entire data content of a database, as described in a conceptual schema. *Note*: There may be many external views of a database, but only one conceptual view. [610.5]

concrete model. A model in which at least one component represented is a tangible object; for example, a physical replica of a building. [610.3]

concurrent. Pertaining to the occurrence of two or more activities within the same interval of time, achieved either by interleaving the activities or by simultaneous execution. *Syn:* **parallel (2).** *Contrast with:* **simultaneous.** [610.12]

concurrent reorganization. Database reorganization in which users have access to the reorganized portion of the database while one or more reorganization processes are modifying other portions of the database. [610.5]

condition. (1) The values assumed at a given instant by the variables in a system, model, or simulation. *See also:* **boundary condition; final condition; initial condition; state.** [610.3]
(2) *See*: **atomic condition; item condition; qualification condition; query condition; record condition.** [610.5]

conditional branch.* *See:* **conditional jump.**
*Deprecated. [610.12]

conditional event. A sequentially dependent event that will occur only if some other event has already taken place. *See also:* **time-dependent event.** [610.3]

conditional implication. *See:* **implication.** [610.1]

conditional jump. A jump that takes place only when specified conditions are met. *Contrast with:* **unconditional jump.** [610.12]

condition code. *See:* **status code.** [610.12]

Conference on Data Systems Languages (CODASYL). The organization that establishes industry standards for database structures. [610.5]

configuration. (1) The arrangement of a computer system or component as defined by the number, nature, and interconnections of its constituent parts. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

(2) In configuration management, the functional and physical characteristics of hardware or software as set forth in technical documentation or achieved in a product. *See also:* **configuration item; form, fit and function; version.** [610.12]

configuration audit. *See:* **functional configuration audit; physical configuration audit.** [610.12]

configuration control. An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification. *Syn:* **change control.** *Contrast with:* **configuration identification; configuration status accounting.** *See also:* **configuration control board; deviation; engineering change; interface control; notice of revision; specification change notice; waiver.** [610.12]

configuration control board (CCB). A group of people responsible for evaluating and approving or disapproving proposed changes to configuration items, and for ensuring implementation of approved changes. *Syn:* **change control board.** *See also:* **configuration control.** [610.12]

configuration diagram. *See:* **block diagram.** [610.12]

configuration identification. (1) An element of configuration management, consisting of selecting the configuration items for a system and recording their functional and physical characteristics in technical documentation. *Contrast with:* **configuration control; configuration status accounting.** [610.12]

(2) The current approved technical documentation for a configuration item as set forth in specifications, drawings, associated lists, and documents referenced therein. *See also:* **allocated configuration identification; functional configuration identification; product configuration identification; baseline.** [610.12]

configuration index. A document used in configuration management, providing an accounting of the configuration items that make up a product. *See also:* **configuration item development record; configuration status accounting.** [610.12]

configuration item (CI). An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process. *See also:* **hardware configuration item; computer software configuration item; configuration identification; critical item.** [610.12]

configuration item development record. A document used in configuration management, describing the development status of a configuration item based on the results of configuration audits and design reviews. *See also:* **configuration index; configuration status accounting.** [610.12]

configuration management (CM). A discipline applying technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements. *See also:* **baseline; configuration identification; configuration control; configuration status accounting; configuration audit.** [610.12]

configuration status accounting. An element of configuration management, consisting of the recording and reporting of information needed to manage a configuration effectively. This information includes a listing of the approved configuration identification, the status of proposed changes to the configuration, and the implementation status of approved changes. *Contrast with:* **configuration control; configuration identification.** *See also:* **configuration index; configuration item development record.** [610.12]

congruencing. An image processing technique in which two images of the same scene are transformed so that the size, shape, position, and orientation of all objects in one image are the same as those in the other image. [610.4]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

conjunction. (1) The logical 'AND' operator. [610.1a, 610.5]
(2) The result of joining two conditions by the logical 'AND' operator. *Contrast with:* **disjunction.** *See also:* **conjunctive query.** [610.5]

conjunctive query. A database query formed by using one of the logical operators 'AND' and 'OR.' [610.5]

connected. Pertaining to a subset of an image, any two points of which can be joined by an arc that is entirely contained within the subset. [610.4]

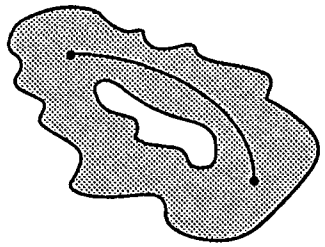


Fig 20
Illustration of Connected

consecutive. Pertaining to the occurrence of two sequential events or items without the intervention of any other event or item; that is, one immediately after the other. [610.12]

consecutive spill method. *See:* **linear probing.** [610.5]

consistency. The degree of uniformity, standardization, and freedom from contradiction among the documents or parts of a system or component. *See also:* **traceability.** [610.12]

consistency check. A check that verifies that an item of data is compatible with certain rules specified for that data. For example, one might wish to check the consistency between two data elements ORDER-DATE and DELIVER-DATE such that DELIVER-DATE may not be earlier than ORDER-DATE. *See also:* **crossfooting check; limit check; range check; validity check.** [610.5]

constant. A quantity or data item whose value cannot change; for example, the data item FIVE, with an unchanging value of 5. *Contrast with:* **variable.** *See also:* **figurative constant; literal.** [610.3a, 610.12]

constant-failure period. The period of time in the life cycle of a system or component during which hardware failures occur at an approximately uniform rate. *Contrast with:* **early-failure period; wearout-failure period.** *See also:* **bathtub curve.** [610.12]

containment. The result of placing all occurrences of a repeating group within the same logical record. [610.5]

content coupling. A type of coupling in which some or all of the contents of one software module are included in the contents of another module. *Contrast with:* **common-environment coupling; control coupling; data coupling; hybrid coupling; pathological coupling.** [610.12]

contents list. In word processing, a list of stored information available for user selection. [610.2]

context editing. A method of line editing in which the line to be viewed or altered is identified by specifying part or all of its contents. [610.2]

contiguous allocation. A storage allocation technique in which programs or data to be stored are allocated a block of storage of equal or greater size, so that logically contiguous programs and data are assigned physically contiguous storage locations. *Contrast with:* **paging (1).** [610.12]

continuous change model. *See:* **continuous model.** [610.3]

continuous iteration. A loop that has no exit. [610.12]

continuous model. (1) A mathematical or computational model whose output variables change in a continuous manner; that is, in changing from one value to another, a variable can take on all intermediate values; for example, a model depicting the rate of air flow over an airplane wing. *Syn:* **continuous-variable model.** [610.3]
(2) A model of a system that behaves in a

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

continuous manner. *Contrast with:* **discrete model.** [610.3]

continuous-progression code. *See:* **unit-distance code.** [610.1]

continuous simulation. A simulation that uses a continuous model. [610.3]

continuous-variable model. *See:* **continuous model.** [610.3]

contour analysis. In optical character recognition, a technique for locating the outline of a character by searching around its exterior edges with a spot of light. [610.2]

contour encoding. An image compression technique in which a region that has a constant gray level is encoded by specifying only its border. [610.4]

contrast. In image processing, the difference between the average brightness of two subsets of an image. [610.4]

contrast stretching. An image enhancement technique in which the contrast between image subsets and their complements is increased. [610.4]

control breakpoint. *See:* **code breakpoint.** [610.12]

control character. (1) A character whose occurrence in a particular context initiates, modifies, or stops a control operation. A control character may be recorded for use in a subsequent action, and it may have a graphic representation in some circumstances. *See also*: **accuracy control character; device control character; print control character; transmission control character.** *Syn*: **command character; functional character; instruction character; operational character.** [610.5]
(2) A character that initiates some kind of physical control action but is not printed on the output page. For example, line feed, tab, form feed. [610.5]

control coupling. A type of coupling in which one software module communicates information to another module for the explicit purpose of influencing the latter module's execution. *Contrast with:* **common-environment coupling; content coupling; data coupling; hybrid coupling; pathological coupling.** [610.12]

control data. Data that select an operating mode, direct the sequential flow of a program, or otherwise directly influence the operation of software; for example, a loop control variable. [610.12]

control flow. The sequence in which operations are performed during the execution of a computer program. *Syn:* **flow of control.** *Contrast with:* **data flow.** [610.12]

control flow diagram. A diagram that depicts the set of all possible sequences in which operations may be performed during the execution of a system or program. Types include box diagram, flowchart, input-process-output chart, state diagram. *Contrast with:* **data flow diagram.** *See also:* **call graph; structure chart.** [610.12]

control flow trace. *See:* **execution trace.** [610.12]

control language. *See:* **job control language.** [610.12]

controlled list. (1) A list whose access is controlled in some way; for example, access to an array is controlled by its index variable. [610.5]
(2) A list that can contain a finite number of entries. [610.5]

controlled list data element. A data element that is contained in a controlled list. [610.5]

control program. *See:* **supervisory program.** [610.12]

control statement. A program statement that selects among alternative sets of program statements or affects the order in which operations are performed. For example, if-then-else, case. *Contrast with:* **assignment statement; declaration.** [610.12]

control store. In a microprogrammed computer, the computer memory in which microprograms reside. *See also:* **microword; nanostore.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

control system. A system in which a desired effect is achieved by operating on inputs until the output, which is a measure of the desired effect, falls within an acceptable range of values. *See also:* **automatic control; closed loop control; open-loop control.** [610.2]

control total. *See:* **hash total.** [610.5]

control variable. *See:* **loop-control variable.** [610.12]

conventions. Requirements employed to prescribe a disciplined uniform approach to providing consistency in a software product, that is, uniform patterns or forms for arranging data. *See also:* **practices; standards.** [610.12]

conversational. Pertaining to an interactive system or mode of operation in which the interaction between the user and the system resembles a human dialog. *Contrast with:* **batch.** *See also:* **interactive; on-line; real time.** [610.12]

conversational compiler. *See:* **incremental compiler.** [610.12]

converse inorder traversal. The process of traversing a binary tree in a recursive fashion as follows: the right subtree is traversed, then the root is visited, then the left subtree is traversed. *Contrast with:* **converse postorder traversal; converse preorder traversal.** *See also:* **inorder traversal.** [610.5]

converse postorder traversal. The process of traversing a binary tree in a recursive fashion as follows: the right subtree is traversed, then the left subtree is traversed, then the root is visited. *Contrast with:* **converse inorder traversal; converse preorder traversal.** *See also:* **postorder traversal.** [610.5]

converse preorder traversal. The process of traversing a binary tree in a recursive fashion as follows: the root is visited, then the right subtree is traversed, then the left subtree is traversed. *Contrast with:* **converse inorder traversal; converse postorder traversal.** *See also:* **preorder traversal.** [610.5]

conversion. Modification of existing software to enable it to operate with similar functional capability in a different environment; for example, converting a program from Fortran to Ada, converting a program that runs on one computer to run on another. [610.12]

convex. Pertaining to a region for which a straight line segment between any two points of the region is entirely contained within the region. *Contrast with:* **concave.** [610.4]

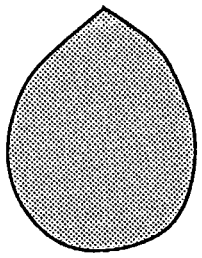


Fig 21
Illustration of Convex

convex programming. In operations research, a particular type of nonlinear programming in which the function to be maximized or minimized and the constraints to be applied are appropriately convex or concave functions, respectively. [610.2]

copy. (1) To read data from a source, leaving the source data unchanged, and to write the same data elsewhere in a physical form that may differ from that of the source. For example, to copy data from a magnetic disk onto a magnetic tape. *Contrast with:* **move.** [610.2, 610.12]
(2) The result of a copy process as in (1). For example, a copy of a data file. *See also:* **display; hard copy; soft copy.** [610.2, 610.12]

core dump.* *See:* **memory dump.** [610.12]
*Deprecated.

coroutine. A routine that begins execution at the point at which operation was last suspended, and that is not required to return control to the program or subprogram that called it. *Contrast with:* **subroutine.** [610.12]

correction. A quantity (equal in absolute value to the error) added to a calculated or observed value to obtain the true value. [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052795 6

corrective maintenance. Maintenance performed to correct faults in hardware or software. *Contrast with:* **adaptive maintenance; perfective maintenance.** [610.12]

correctness. (1) The degree to which a system or component is free from faults in its specification, design, and implementation. [610.12]
(2) The degree to which software, documentation, or other items meet specified requirements. [610.12]
(3) The degree to which software, documentation, or other items meet user needs and expectations, whether specified or not.[610.12]

correctness proof. *See:* **proof of correctness.** [610.12]

correspondence. *See:* **fidelity.** [610.3]

counter. A variable used to record the number of occurrences of a given event during the execution of a computer program; for example, a variable that records the number of times a loop is executed. [610.12]

coupling. The manner and degree of interdependence between software modules. Types include common-environment coupling, content coupling, control coupling, data coupling, hybrid coupling, and pathological coupling. *Contrast with:* **cohesion.** [610.12]

courseware. Instructional materials, such as software and student documentation, designed for use in computer-based instruction. [610.2]

CPC. Acronym for **computer program component.** *See:* **computer software component.** [610.12]

CPCI. Acronym for **computer program configuration item.** *See:* **computer software configuration item.** [610.12]

CPM. Acronym for **critical path method.** [610.2]

CR. (1) Acronym for the **carriage return character.** [610.5]
(2) Abbreviation denoting the symbol for "credit." [610.5]

crash. The sudden and complete failure of a computer system or component. *See also:* **hard failure.** [610.12]

critical design review (CDR). (1) A review conducted to verify that the detailed design of one or more configuration items satisfy specified requirements; to establish the compatibility among the configuration items and other items of equipment, facilities, software, and personnel; to assess risk areas for each configuration item; and, as applicable, to assess the results of producibility analyses, review preliminary hardware product specifications, evaluate preliminary test planning, and evaluate the adequacy of preliminary operation and support documents. *See also:* **preliminary design review; system design review.** [610.12]
(2) A review as in (1) of any hardware or software component. [610.12]

critical event simulation. A simulation that is terminated by the occurrence of a certain event; for example, a model depicting the year-by-year forces leading up to a volcanic eruption, that is terminated when the volcano in the model erupts. *See also:* **time-slice simulation.** [610.3]

critical item. In configuration management, an item within a configuration item that, because of special engineering or logistic considerations, requires an approved specification to establish technical or inventory control at the component level. [610.12]

criticality. The degree of impact that a requirement, module, error, fault, failure, or other item has on the development or operation of a system. *Syn:* **severity.** [610.12]

critical path. In the critical path method, a path whose sum of activity times is greater than or equal to the sum of activity times for any other path through the network. *Note:* This sum of activity times is the shortest possible completion time of the overall project. [610.2]

critical path method (CPM). A project management technique in which the activities that constitute a project are identified, dependencies among the activities are determined, a network of parallel and sequen-

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052796 8

tial activities is produced, an estimated time is assigned to each activity, and a sequence of activities taking the longest time (a critical path) is identified, determining the shortest possible completion time for the overall project. *See also:* **project evaluation and review technique.** [610.2]

critical piece first. A system development approach in which the most critical aspects of a system are implemented first. The critical piece may be defined in terms of services provided, degree of risk, difficulty, or other criteria. *See also:* **bottom-up; top-down.** [610.12]

critical software. Software whose failure could have an impact on safety, or could cause large financial or social loss. [610.12]

cross-assembler. An assembler that executes on one computer but generates machine code for a different computer. [610.12]

cross check. To test for accuracy by comparing the results of two different methods of computation. [610.1]

cross-compiler. A compiler that executes on one computer but generates machine code for a different computer. [610.12]

crossfooting check. A consistency check in which two totals obtained by adding the same set of numbers in different sequences are compared. [610.5]

cross-index. A link between two files containing related data. For example, in a library, the subject card catalog is a cross-index for the title and author card catalogs. *See also*: **cross-indexed file; inverted file.** [610.5]

cross-indexed file. A file whose contents are linked with another file through a cross-index. *See also*: **inverted file.** [610.5]

cross-indexing. (1) The process of linking entities in two files to facilitate searches performed on data contained in those files. *See also*: **cross-indexed file.** [610.5] (2) A method of linking entities as in (1). [610.5]

cross-reference generator. A software tool that accepts as input the source code of a computer program and produces as output a listing that identifies each of the program's variables, labels, and other identifiers and indicates which statements in the program define, set, or use each one. *Syn:* **cross-referencer.** [610.12]

cross-reference list. A list that identifies each of the variables, labels, and other identifiers in a computer program and indicates which statements in the program define, set, or use each one. [610.12]

cross-referencer. *See:* **cross-reference generator.** [610.12]

cross-section. In image processing, the intersection of an n-dimensional image or region with an $(n-1)$-dimensional object. For example, the intersection of a two-dimensional image or region with a straight line. [610.4]

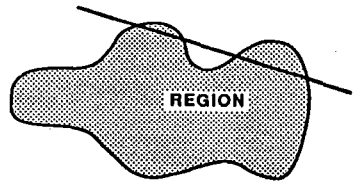


Fig 22
Illustration of Cross Section

cryogenics. A branch of technology concerned with devices that make use of the properties assumed by materials at temperatures near absolute zero. [610.2]

CSC. Acronym for **computer software component.** [610.12]

CSCI. Acronym for **computer software configuration item.** [610.12]

CT. Acronym for **computed tomography.** [610.2]

CTS. Acronym for **computer typesetting.** *See:* **computer-aided typesetting.** [610.2]

cue. *See:* **call (3).** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

currency symbol character. A character within a picture specification that represents the currency sign. *Note*: $ is commonly used. [610.5]

curriculum standard. A standard that describes the characteristics of a course of study on a body of knowledge that is offered by an educational institution. [610.12]

cursor. A moveable icon or spot of light on the screen of a display device that indicates the currently selected object or character. [610.2]

cursor control. The ability to modify the position of a cursor by explicit commands.[610.2]

curve. (1) The path traced by a point moving continuously in space. [610.4]
(2) A finite set of pixels representing a path as in (1). *See also:* **arc; closed curve.** [610.4]



Fig 23
Illustrations of Curve

customer-bank communication terminal (CBCT). *See:* **automated teller machine.** [610.2]

cybernetics. A branch of technology concerned with the comparative study of communication and control in living organisms and in machines. [610.2]

cycle. (1) A period of time during which a set of events is completed. *See also:* **software development cycle; software life cycle.** [610.12]
(2) A set of operations that is repeated regularly in the same sequence, possibly with variations in each repetition; for example, a computer's read cycle. *See also:* **pass.** [610.12]

cycle stealing. The process of suspending the operation of a central processing unit for one or more cycles to permit the occurrence of other operations, such as transferring data from main memory in response to an output request from an input/output controller. [610.12]

cyclic binary code. *See:* **Gray code.** [610.1]

cyclic code. *See:* **Gray code.** [610.1]

cyclic decimal code. A binary code in which sequential decimal digits are represented by four-bit BCD expressions, each of which differs from the preceding expression in one place only. *Note:* This is an example of unit-distance code. [610.1]

cyclic permuted code. *See:* **unit-distance code.** [610.1]

cyclic search. A storage allocation technique in which each search for a suitable block of storage begins with the block following the one last allocated. [610.12]

cyclic shift. *See:* **circular shift.** [610.1]

DA. (1) Acronym for **design automation.** [610.2]
(2) Acronym for **data administrator.** [610.5]

daily cycle. One complete execution of a data processing function that must be performed once a day. *See also:* **annual cycle; monthly cycle; quarterly cycle; weekly cycle.** [610.2]

DAM. Acronym for **direct access method.** [610.5]

data. (1) A representation of facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means. *Note:* "Data" is plural for datum, but is often used as a collective noun, as in "The data is in this file." *See also*: **data type; logical data; null data; numeric data; pointer data.** [610.5, 610.12]
(2) Anything observed in the documentation or operation of software that deviates from expectations based on previously verified software products or reference documents. [610.5, 610.12]
(3) Sometimes used as a synonym for documentation. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

data abstraction. (1) The process of extracting the essential characteristics of data by defining data types and their associated functional characteristics and disregarding representation details. *See also:* **encapsulation; information hiding.** [610.12]
(2) The result of the process in (1). [610.12]

data administrator (DA). An individual who is responsible for the definition, organization, supervision, and protection of data within some organization. *See also*: **database administrator.** [610.5]

data aggregate. A collection of two or more data items that are treated as a unit. *See also*: **composite data element.** *Syn*: **aggregate; group item.** [610.5]

data attribute. A characteristic of a unit of data. [610.5]

data bank. (1) A collection of data libraries. *Note*: A record contains one or more items, a file contains one or more records, a library contains one or more files, and a data bank contains one or more libraries. [610.5]
(2) A collection of data relating to a particular subject area. *Note*: The data may or may not be machine-readable. [610.5]

database. (1) A collection of logically related data stored together in one or more computerized files. *Note:* Each data item is identified by one or more keys. *See also*: **database management system.** [610.5a, 610.12a]
(2) In CODASYL, the collection of all the record occurrences, set occurrences, and areas controlled by a specific schema. [610.5]

Database Task Group (DBTG). A task group of the CODASYL Programming Language Committee that established a set of standards for specification and design of network database structures. *See also*: **CODASYL database.** [610.5]

database access method. A technique for organizing and storing a physical database in computer storage. [610.5]

database administration (DBA). The responsibility for the definition, operation, protection, performance, and recovery of a database. [610.5]

database administrator (DBA). An individual who is responsible for the definition, operation, protection, performance, and recovery of a database. *See also*: **data administrator.** [610.5]

database command language (DBCL). A procedural data manipulation language used to access a database through a database management system. [610.5]

database creation. The process of naming, allocating space, formatting, and defining a database. *See also*: **database definition; database design.** [610.5]

database definition. (1) The process of translating a conceptual schema for a database into a data storage schema. *See also*: **database creation; database design; redefinition.** [610.5]
(2) The result of such a translation. [610.5]

database description language (DBDL). *See*: **data definition language.** [610.5]

database descriptive language. *See*: **data definition language.** [610.5]

database design. (1) The process of developing a conceptual schema for a database that will meet a user's requirements. *See also*: **database creation; database definition.** *Syn*: **implementation design.** [610.5]
(2) The result of the process in (1). [610.5]

database extract. A file, each record of which contains data items selected from a database based on a particular criterion. [610.5]

database integrity. The degree to which the data in a database are current, consistent and accurate. *See also*: **data integrity; database security; integrity.** [610.5]

database key. A field in a database that identifies a record in that database. [610.5]

database management system (DBMS). A computer system involving hardware, software, or both that provides a systematic approach to creating, storing, retrieving and

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

processing information stored in a database. A DBMS acts as an interface between computers' programs and data files as well as between users and the database. It may include backup/recovery, checkpoint processing, and ad-hoc query capability. [610.5]

database manipulation language (DBML). *See*: **data manipulation language.** [610.5]

database organization. The manner in which a database is structured; for example, a hierarchical organization, a relational organization. *See also*: **reorganization.** [610.5]

database record. (1) A collection of data elements that are stored in a database.[610.5] (2) A collection of hierarchically dependent segments (one root and all its descendants) within a hierarchical database. *See also*: **record.** [610.5]

database reorganization. *See*: **reorganization.** [610.5]

database security. The degree to which a database is protected from exposure to accidental or malicious alteration or destruction. *See also*: **data security; database integrity.** [610.5]

database segment. *See*: **segment.** [610.5]

database sublanguage. *See*: **data sublanguage.** [610.5]

database system. A software system that supports multiple applications using a common database. [610.5]

data block. *See*: **block.** [610.5]

data-break. *See*: **direct memory access.** [610.5]

data breakpoint. A breakpoint that is initiated when a specified data item is accessed. *Syn:* **storage breakpoint.** *Contrast with:* **code breakpoint.** *See also:* **dynamic breakpoint; epilog breakpoint; programmable breakpoint; prolog breakpoint; static breakpoint.** [610.12]

data carrier. Material that serves as a data medium or to which a data medium is applied and that facilitates the transport of data; for example, a punch card, a disk, or a plastic card with a magnetic surface that serves as the data medium. *See also*: **data medium.** [610.5]

data chain. *See*: **composite data element.**[610.5]

data characteristic. An inherent, possibly accidental, trait, quality, or property of data (for example, arrival rates, formats, value ranges, or relationships between field values). [610.12]

data code. *See*: **code.** [610.5]

data compaction. Any technique used to encode data in order to reduce the amount of storage it requires. *Contrast with*: **data compression.** [610.5]

data compression. Any technique used to reduce the amount of storage required to store data. *Contrast with*: **data compaction.** [610.5]

data conversion. To change data from one form of representation to another; for example, to convert data from an ASCII representation to an EBCDIC representation. [610.5]

data coupling. A type of coupling in which output from one software module serves as input to another module. *Syn:* **input-output coupling.** *Contrast with:* **common-environment coupling; content coupling; control coupling; hybrid coupling; pathological coupling.** [610.12]

data definition. A description of the format, structure, and properties of a data item, data element, or data structure. [610.5]

data definition language (DDL). (1) A language for describing the organization of data within a database. *Note*: In some software, the logical organization is described; in some, both the logical and physical organizations are described. [610.5] (2) A language used to describe the logical structure of a database. *Syn*: **data description language; database description language; database descriptive language; schema definition language; schema language.** *Contrast with:* **data manipulation**

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

language. *See also:* **database manipulation language; subschema data definition language.** [610.5]

data density. The amount of data that can be stored in one unit of data medium. For example, the number of bits stored in an inch of magnetic tape medium. [610.5]

data description language. *See*: **data definition language.** [610.5]

data dictionary. A collection of entries specifying the name, source, usage, and format of each data element used in a system or set of systems. *See also*: **data directory.** *Syn*: **data element dictionary.** [610.5]

data dictionary/directory (DD/D). *See*: **data dictionary; data directory.** [610.5]

data dictionary system. A software system that maintains and manages a data dictionary. [610.5]

data directory. A collection of entries specifying the data name, source, location, ownership, usage and format of each data element used in some system or set of systems. *See also*: **data dictionary.** [610.5]

data element. (1) A uniquely named and defined component of a data definition; a data "cell" into which data items (actual values) can be placed. For example, the data element AGE, into which data items 1, 2, ... can be placed. *Note*: The terms **data element** and **data item** are often used interchangeably or with the reverse definitions from those given here. No standard of use exists at this time. *Syn*: **cell.** [610.5]
(2) A data definition as in (1) that cannot be divided into other individually named data definitions. *See also*: **attribute; data item.** [610.5]

data element dictionary. *See*: **data dictionary.** [610.5]

data element tag. *See*: **code.** [610.5]

data entry. To input data into a computer system. [610.5]

data exception. An exception that occurs when a program attempts to use or access data incorrectly. *See also:* **addressing exception; operation exception; overflow exception; protection exception; underflow exception.** [610.12]

data exchange. (1) The use of data by more than one computer program or system. *See also*: **data interchange; exchange data.** [610.5]
(2) The movement of data between two or more programs or systems. [610.5]

data field. *See*: **attribute.** [610.5]

data file. *See*: **file.** [610.5]

data flow. The sequence in which data transfer, use, and transformation are performed during the execution of a computer program. *Contrast with:* **control flow.** [610.12]

data flowchart (flow chart). *See:* **data flow diagram.** [610.12]

data flow diagram (DFD). A diagram that depicts data sources, data sinks, data storage, and processes performed on data as nodes, and logical flow of data as links between the nodes. *Syn:* **data flowchart; data flow graph.** *Contrast with:* **control flow diagram; data structure diagram.** [610.12]

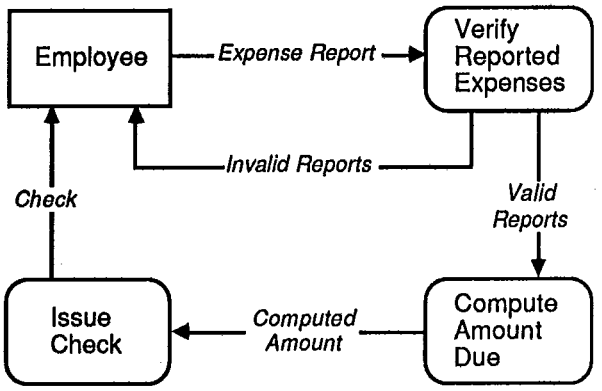


Fig 24
Data Flow Diagram [610.12]

data flow graph. *See:* **data flow diagram.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

data flow trace. *See:* **variable trace.** [610.12]

data glossary. A collection of entries specifying a data definition and a specification of its uses. [610.5]

data hierarchy. A set of directed relationships between two or more units of data, such that each unit has one and only one owner. *See also*: **hierarchy.** [610.5]

data independence. The degree to which the logical view of a database is immune to changes in the physical structure of the database. [610.5]

data input sheet. User documentation that describes, in a worksheet format, the required and optional input data for a system or component. *See also:* **user manual.** [610.12]

data integrity. The degree to which a collection of data is complete, consistent, and accurate. *See also*: **data security; database integrity; integrity.** *Syn*: **data quality.** [610.5]

data interchange. The use of data by two or more different systems. *See also*: **data exchange.** [610.5]

data interchange format (DIF). A standardized data file format allowing data interchange between software packages on personal computers. For example, data interchange between an electronic spread sheet and a word processor could be accomplished by converting the spread sheet data to data interchange format, then to the format required for the word processor. [610.2]

data item. A value contained in a data element; for example the data element AGE might contain data items 1, 2, ... *Note*: The terms **data element** and **data item** are often used interchangeably or with the reverse definitions from those given here. No standard of use exists at this time. *See also*: **data element.** [610.5]

data library. A set of related files, tables, or sets. [610.5]

data link. The physical means of connecting two computers together for the purpose of transmitting and receiving data. [610.5]

data management. The function of controlling the acquisition, analysis, storage, retrieval, and distribution of data. [610.5]

data manipulation language (DML). A language used to retrieve, insert, delete, or modify the data in a database. *Syn*: **database manipulation language.** *Contrast with*: **data definition language.** [610.5]

data medium. The material in or on which data is or may be represented. *See also*: **data carrier.** [610.5]

data model. A description of data that consists of all entities represented in a data structure or database and the relationships that exist among them. *See also*: **logical data model; physical data model; schema; view.** [610.5]

data name. One or more characters used to identify a data element. [610.5]

data normalization. *See*: **normalization.** [610.5]

data processing (DP). The systematic performance of operations upon data, such as data manipulation, merging, sorting, and computing. *Syn:* **information processing.** *See also:* **administrative data processing; automatic data processing; business data processing; commercial data processing; distributed data processing; integrated data processing; mechanical data processing; office automation; remote-access data processing.** [610.2]

data processing cycle. *See:* **processing cycle.** [610.2]

data processing system. A system, including computer systems and associated personnel, that performs input, processing, storage, output, and control functions to accomplish a sequence of operations on data. [610.2]

data quality. *See*: **data integrity.** [610.5]

data record. *See*: **record.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

data reduction. Any technique used to transform data from raw data into a more useful form of data. For example, grouping, summing, or averaging related data. [610.5]

data resource. A purposely organized body of data that is of use to some person or group of people. [610.5]

data security. The degree to which a collection of data is protected from exposure to accidental or malicious alteration or destruction. *See also*: **data integrity; database security.** [610.5]

data-sensitive fault. A fault that causes a failure in response to some particular pattern of data. *Syn:* **pattern-sensitive fault.** *Contrast with:* **program-sensitive fault.** [610.12]

data set. A named collection of related records. *Note*: Often synonymous with file. *See also*: **partitioned data set.** [610.5]

data storage description language. A language used to define the organization of stored data in terms that are independent of any particular storage device or operating system. [610.5]

data storage schema. A data structure that describes the manner in which data items are physically stored in storage. *See also*: **database definition.** [610.5]

data structure. A physical or logical relationship among data elements, designed to support specific data manipulation functions. *Syn*: **logical structure.** [610.5a, 610.12]

data structure-centered design. A software design technique in which the architecture of a system is derived from analysis of the structure of the data sets with which the system must deal. *See also:* **input-process-output; modular decomposition; object-oriented design; rapid prototyping; stepwise refinement; structure clash; structured design; transaction analysis; transform analysis.** [610.12]

data structure diagram. A diagram that depicts a set of data elements, their attributes, and the logical relationships among them. *Contrast with:* **data flow diagram.** *See also:* **entity-relationship diagram.** [610.12]

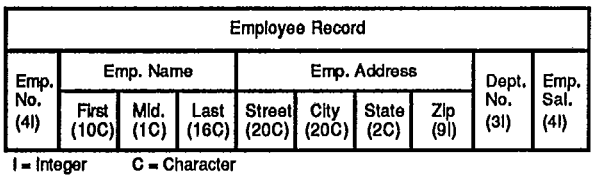


Fig 25
Data Structure Diagram

data sublanguage (DSL). A subset of another language, called the host language, that is used to perform database operations. *Syn*: **database sublanguage.** [610.5]

data submodel. *See*: **external schema.** [610.5]

data trace. *See:* **variable trace.** [610.12]

data translation. The modification of the physical representation of data used in one hardware/software environment so that it is compatible with a different hardware/software environment. [610.5]

data type. A class of data, characterized by the members of the class and the operations that can be applied to them. For example, character type, enumeration type, integer type, logical type, real type. *See also:* **strong typing.** [610.12]

data value. The actual value that is stored in a data item. For example, the numeric value of the data item SALARY may be 20,000. *Syn*: **value.** [610.5]

data volatility. The rate of change, over a specified period of time, in the values of stored data items. [610.5]

datum. Singular for **data.** [610.5, 610.12]

daughter. *See*: **child node.** [610.5]

DBA. Acronym for **database administration** or **database administrator.** [610.5]

DBCL. Acronym for **database command language.** *See*: **database manipulation language.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

DBDL. Acronym for **database description language.** *See*: **data definition language.** [610.5]

DBML. Acronym for **database manipulation language.** [610.5]

DBMS. Acronym for **database management system.** [610.5]

DbNC. Acronym for **distributed numerical control.** [610.2]

DBTG. Acronym for **Database Task Group.** [610.5]

DD/D. Acronym for **data dictionary/directory.** [610.5]

DDL. Acronym for **data definition language.** [610.5]

DDP. (1) Acronym for **distributed data processing.** [610.2]
(2) Acronym for **dispersed data processing.** *See:* **distributed data processing.** [610.2]

deadlock. A situation in which computer processing is suspended because two or more devices or processes are each awaiting resources assigned to the other(s). For example, a situation in which computer program A, with an exclusive lock on record X, asks for a lock on record Y, which is allocated to computer program B. Likewise, program B is waiting for exclusive control over record X before giving up control of record Y. *Syn:* **deadly embrace.** *See also:* **lockout.** [610.5, 610.12]

deadly embrace. *See*: **deadlock.** [610.5]

deassembler.* *See:* **disassembler.** [610.12]
* Deprecated.

deblock. To separate the parts of a block. *Contrast with:* **block (2).** [610.12]

deblock. To separate the parts of a block. *Syn*: **unblock.** *Contrast with*: **block (2).** [610.5a, 610.12]

deblurring. *See:* **sharpening.** [610.4]

debug. To detect, locate, and correct faults in a computer program. Techniques include use of breakpoints, desk checking, dumps, inspection, reversible execution, single-step operation, and traces. [610.12]

deca (deka). A prefix indicating ten. [610.1]

deci. A prefix indicating one tenth. [610.1]

decimal. (1) Pertaining to a selection in which there are ten possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of ten. *Syn:* **denary.** [610.1]

decimal alignment. An operation in which two or more decimal numbers are arranged such that their decimal points, whether explicit or implicit, are aligned vertically; for example,

unaligned	aligned
1.4	1.4
5	5
.067	.067

See also: **radix alignment.** [610.2a, 610.5a]

decimal-coded digit. A digit or character defined by a set of decimal digits, such as a pair of decimal digits specifying a letter or special character in a system of notation. [610.1]

decimal data. Data used to represent decimal numbers; that is, numeric values expressed in base 10. *See also*: **binary coded decimal real data; decimal picture data; packed decimal data; unsigned packed decimal data; zoned decimal data.** [610.5]

decimal digit. A numeral used to represent one of the ten digits in the decimal numeration system; 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9. [610.1]

decimal notation. Any notation that uses the decimal digits and the radix 10. [610.1]

decimal number. (1) A quantity that is expressed using the decimal numeration system. [610.1]
(2) Loosely, a decimal numeral. [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

decimal number system.* *See:* **decimal numeration system.** [610.1]
*Deprecated.

decimal numeral. A numeral in the decimal numeration system. For example, the decimal numeral 12 is equivalent to the Roman numeral XII. [610.1]

decimal numeration system. The numeration system that uses the decimal digits and the radix 10. *Syn:* **decimal system.** [610.1]

decimal picture data. Arithmetic picture data that is associated with a picture specification that allows decimal digit characters, a radix point, zero-suppression characters, sign characters, currency symbol characters, insertion characters, commercial characters, and exponent characters. *Syn:* **numeric character data.** *Contrast with:* **binary picture data.** [610.5]

decimal point. The radix point in the decimal numeration system. [610.1]

decimal system. *See:* **decimal numeration system.** [610.1]

decimal-to-BCD conversion. The process of converting a decimal numeral to an equivalent BCD numeral. For example, decimal 139 is converted to BCD 1 0011 1001. [610.1]

decimal-to-binary conversion. The process of converting a decimal numeral to an equivalent binary numeral. For example, decimal 139.25 is converted to binary 10001011.01. [610.1]

decimal-to-hexadecimal conversion. The process of converting a decimal numeral to an equivalent hexadecimal numeral. For example, decimal 139.25 is converted to hexadecimal 8B.4. [610.1]

decimal-to-octal conversion. The process of converting a decimal numeral to an equivalent octal numeral. For example, decimal 139.25 is converted to octal 213.2. [610.1]

decision rule. A rule or algorithm used in pattern classification to assign an observed unit of image data to a pattern class based on features extracted from the image. *Syn:* **classifier.** [610.4]

decision support services (DSS). (1) The services provided by a decision support system. [610.2]
(2) The services provided by the staff of an information center. [610.2]

decision support software. Interactive software used in a decision support system. For example, software components for model building, forecasting, statistical analysis, ad hoc model interrogation, report generation, and graphics. [610.2]

decision support system (DSS). A computer system that supports decision making by performing such functions as modeling, forecasting, and statistical analysis. *See also:* **computer-aided management; management information system.** [610.2]

decision support system generator. A package of decision support software that enables users to develop customized decision support systems for specific applications. [610.2]

decision table. A table used to show sets of conditions and the actions resulting from them. [610.12]

declaration. A non-executable program statement that affects the assembler or compiler's interpretation of other statements in the program. For example, a statement that identifies a name, specifies what the name represents, and, possibly, assigns it an initial value. *Contrast with:* **assignment statement; control statement.** *See also:* **pseudo instruction.** [610.12]

declarative language. A nonprocedural language that permits the user to declare a set of facts and to express queries or problems that use these facts. *See also:* **interactive language; rule-based language.** [610.12]

decode. To convert data by reversing the effect of previous encoding. *Contrast with:* **encode.** [610.5]

decoder. A device or system that decodes data. *Contrast with:* **encoder.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

decollate. To divide the items in a set into unique subsets. *Contrast with*: **collate.**[610.5]

decompile. To translate a compiled computer program from its machine language version into a form that resembles, but may not be identical to, the original high order language program. *Contrast with:* **compile.** [610.12]

decompiler. A software tool that decompiles computer programs. [610.12]

decoupling. The process of making software modules more independent of one another to decrease the impact of changes to, and errors in, the individual modules. *See also:* **coupling.** [610.12]

decrement. (1) The quantity by which a variable is decreased. [610.1]
(2) To decrease the value of a variable. *Contrast with:* **increment (2).** [610.1]
(3) To decrease the value of a variable by one. *Contrast with:* **increment (3).** [610.1]

dedicated word processing. Word processing performed on a system used exclusively for that purpose. *Contrast with:* **clustered word processing; shared-logic word processing; shared-resource word processing; stand-alone word processing.** [610.2]

default. (1) Pertaining to a value, attribute, or option that is assumed in place of a value, attribute or option when one is required, but not specified explicitly; for example, the default value for a field called MARITAL STATUS might be M (for married). [610.5]
(2) To assign the value, attribute, or option as in (1). [610.5]

definition. *See*: **database definition.** [610.5]

degenerate tree. A tree in which each nonterminal node has exactly one subtree.[610.5]

degree. (1) With regard to a relation, the number of attributes in that relation. [610.5]
(2) With regard to a given node in a tree, the number of subtrees within that node. [610.5]

deka. *See:* **deca.** [610.1]

DEL. Abbreviation for the **delete character.** [610.5]

delete. (1) To remove data from a storage device or data medium. *See also*: **read; update; write.** *Syn*: **erase; physical delete.** [610.5]
(2) To render data unretrievable, although it may continue to be physically present on a storage device. *Syn*: **logical delete.** [610.5]

delete access. A type of access to data in which the data may be deleted. *See also*: **read-only access; read/write access; update access; write access.** [610.5]

delete character (DEL). A control character used to obliterate an erroneous or unwanted character. *Note*: On a perforated tape, this character consists of a card hole in each punch position. *Syn*: **erase character; rub-out character.** [610.5]

delete transaction. A transaction that causes a record to be deleted from a master file. *See also:* **add transaction; change transaction; null transaction; update transaction.** [610.2]

deletion record. (1) A record that indicates that data is to be deleted. [610.5]
(2) A record that has been deleted from a master file. *Contrast with*: **add record.**[610.5]

delimiter. A bit, character, or set of characters used to denote the beginning or end of a group of related bits, characters, words, or statements. For example, the ampersand "&" in the character string "&APPLE&." *Syn*: **separator.** [610.5, 610.12a]

delink. To retrieve and delete an item from a linked list. [610.5]

delivery. Release of a system or component to its customer or intended user. *See also:* **software life cycle; system life cycle.**[610.12]

demand paging. A storage allocation technique in which pages are transferred from auxiliary storage to main storage only when those pages are needed. *Contrast with:* **anticipatory paging.** [610.12]

demodularization. In software design, the process of combining related software mod-

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052806 7

ules, usually to optimize system performance. *See also:* **downward compression; lateral compression; upward compression.** [610.12]

demonstration. A dynamic analysis technique that relies on observation of system or component behavior during execution, without need for post-execution analysis, to detect errors, violations of development standards, and other problems. *See also:* **testing.**[610.12]

denary. *See:* **decimal (2).** [610.1]

denormalized number. A nonzero floating-point number whose exponent has a reserved value, usually the format's minimum, and whose explicit or implicit significand digit is zero. [610.1]

dense binary code. A binary code in which all possible bit combinations are used. [610.1]

dense list. *See*: **packed array.** [610.5]

dependency. *See*: **functional dependency; join dependency; multivalued dependency; nontransitive dependency; transitive dependency.** [610.5]

dependent segment. In a hierarchical database, a segment that is not the root segment. [610.5]

dependent variable. A variable whose value is dependent on the values of one or more independent variables. *Contrast with:* **independent variable.** [610.3]

depth. *See*: **height.** [610.5]

depth-first search (DFS). A search of a tree using preorder traversal. [610.5]

deque. Acronym for **double-ended queue.** Pronounced "deck." [610.5]

dequeue. To retrieve and delete an item from a queue. *Contrast with*: **enqueue.** [610.5]

derivative indexing. Automatic indexing in which the keywords are extracted directly from the text of the document or information being indexed. For example, keyword in context index. *Syn:* **derived-term indexing; extraction indexing.** *Contrast with:* **assigned indexing.** *See also:* **uniterm indexing.** [610.2]

derived data. Data that is computed or otherwise obtained from other data by application of a specified procedure. [610.5]

derived data element. A data element whose entries are obtained from those in another data element by application of a specified procedure; for example, entries in a data element "age" could be derived from entries in the data element "date of birth." [610.5]

derived relation. A relation in a database that can be entirely obtained from previously defined base relations by applying some sequence of relational operators. *Contrast with*: **base relation.** [610.5]

derived type. A data type whose members and operations are taken from those of another data type according to some specified rule. *See also:* **subtype.** [610.12]

derived-term indexing. *See:* **derivative indexing.** [610.2]

descendant node. In a tree, a node that is in a subtree of a given node. *See also*: **ancestor; parent node; root note.** [610.5]

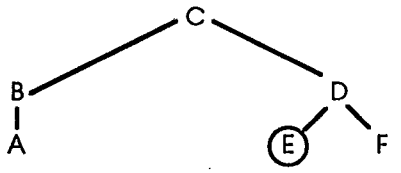


Fig 26
Node E Is a Descendant Node of Nodes C and D

descender. The portion of a graphic character that extends below the main part of the character; for example the lower portion of the letters "g" and "y." *Contrast with:* **ascender.** [610.2]

description standard. A standard that describes the characteristics of product information or procedures provided to help un-

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

derstand, test, install, operate, or maintain the product. [610.122]

descriptive model. A model used to depict the behavior or properties of an existing system or type of system; for example, a scale model or written specification used to convey to potential buyers the physical and performance characteristics of a computer. *Syn:* **representational model.** *Contrast with:* **prescriptive model.** [610.3]

descriptor. *See:* **keyword.** [610.2]

design. (1) The process of defining the architecture, components, interfaces, and other characteristics of a system or component. *See also:* **architectural design; database design; detailed design; preliminary design.** [610.5, 610.12]
(2) The result of the process in (1). [610.12]

design automation. The use of computers to automate the design process. *See also:* **computer-aided design.** [610.2]

design description. A document that describes the design of a system or component. Typical contents include system or component architecture, control logic, data structures, input/output formats, interface descriptions, and algorithms. *Syn:* **design document; design specification.** *See also:* **product specification.** *Contrast with:* **requirements specification.** [610.12]

design document. *See:* **design description.** [610.12]

design element. A basic component or building block in a design. [610.12]

design entity. An element (component) of a design that is structurally and functionally distinct from other elements and that is separately named and referenced. [610.12]

design inspection. *See:* **inspection.** [610.12]

design language. A specification language with special constructs and, sometimes, verification protocols, used to develop, analyze, and document a hardware or software design. Types include hardware design language, program design language. *See also:* **requirements specification language.** [610.12]

design level. The design decomposition of the software item (for example, system, subsystem, program, or module). [610.12]

design phase. The period of time in the software life cycle during which the designs for architecture, software components, interfaces, and data are created, documented, and verified to satisfy requirements. *See also:* **detailed design; preliminary design.** [610.12]

design requirement. A requirement that specifies or constrains the design of a system or system component. *Contrast with:* **functional requirement; implementation requirement; interface requirement; performance requirement; physical requirement.** [610.12]

design review. A process or meeting during which a system, hardware, or software design is presented to project personnel, managers, users, customers, or other interested parties for comment or approval. Types include critical design review, preliminary design review, system design review. *Contrast with:* **code review; formal qualification review; requirements review; test readiness review.** [610.12]

design specification. *See:* **design description.** [610.12]

design standard. A standard that describes the characteristics of a design or a design description of data or program components. [610.122]

design unit. A logically related collection of design elements. In an Ada PDL, a design unit is represented by an Ada compilation unit. [610.12]

design view. A subset of design entity attribute information that is specifically suited to the needs of a software project activity. [610.12]

designation number. In logic design, the bottom line of a truth table written such that the

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

values of the variables equal the binary number of the state. For example, the designation number for a two-variable exclusive-OR function is as follows:

State	0	1	2	3	
Variable A	0	1	0	1	
Variable B	0	0	1	1	
Exclusive OR	0	1	1	0	(Designation number)

[610.1]

desk checking. A static analysis technique in which code listings, test results, or other documentation are visually examined, usually by the person who generated them, to identify errors, violations of development standards, or other problems. *See also:* **inspection; walk-through.** [610.12]

desktop computer. A computer designed for use on a desk or table. [610.2]

destination. The location for the result of a binary or unary operation. [610.1,]

destination address. The address of the device or storage location to which data is to be transferred. *Contrast with:* **source address.** [610.12]

destructive addition. Computer addition in which the sum is placed in the storage location, register, or accumulator previously occupied by an operand, usually the augend, which is then lost. *Contrast with:* **nondestructive addition.** [610.1]

destructive backspace. In word processing, an operation that moves the cursor back one character and deletes the character that was in the cursor's new location. [610.2]

destructive read. A read operation that alters the data in the accessed location. *Contrast with:* **nondestructive read.** [610.12]

detail file. *See:* **transaction file.** [610.2]

detailed design. (1) The process of refining and expanding the preliminary design of a system or component to the extent that the design is sufficiently complete to be implemented. *See also:* **software development process.** [610.12]
(2) The result of the process in (1). [610.12]

determinant. Within a relation, an attribute on which some other attribute is functionally dependent. [610.5]

deterministic. Pertaining to a process, model, or variable whose outcome, result, or value does not depend on chance. *Contrast with:* **stochastic.** [610.3]

deterministic model. A model in which the results are determined through known relationships among the states and events, and in which a given input will always produce the same output; for example, a model depicting a known chemical reaction. *Contrast with:* **stochastic model.** [610.3]

development cycle. *See:* **software development cycle.** [610.12]

development life cycle. *See:* **software development cycle.** [610.12]

development specification. *See:* **requirements specification.** [610.12]

development testing. Formal or informal testing conducted during the development of a system or component, usually in the development environment by the developer. *Contrast with:* **acceptance testing; operational testing.** *See also:* **qualification testing.** [610.12]

developmental baseline.* *See:* **developmental configuration.** [610.12]
*Deprecated.

developmental configuration. In configuration management, the software and associated technical documentation that define the evolving configuration of a computer software configuration item during development. *Note:* The developmental configuration is under the developer's control, and therefore is not called a baseline. *Contrast with:* **allocated baseline; functional baseline; product baseline.** [610.12]

deviation. (1) A departure from a specified requirement. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

(2) A written authorization, granted prior to the manufacture of an item, to depart from a particular performance or design requirement for a specific number of units or a specific period of time. *Note:* Unlike an engineering change, a deviation does not require revision of the documentation defining the affected item. *See also:* **configuration control.** *Contrast with:* **engineering change; waiver.** [610.12]

device. A mechanism or piece of equipment designed to serve a purpose or perform a function. [610.12]

device control character. A control character used for the control of auxiliary devices associated with a data processing system or data communication system; for example, a control character for switching such devices on or off. [610.5]

device media control language. A language that may be used to describe the physical layout and organization of data within some physical storage media. [610.5]

DFD. Acronym for **data flow diagram.** [610.12]

DFS. Acronym for **depth-first search.** [610.5]

diad. A group of two closely related items or digits. [610.1]

diagnostic. Pertaining to the detection and isolation of faults or failures; for example, a diagnostic message, a diagnostic manual. [610.12]

diagnostic manual. A document that presents the information necessary to execute diagnostic procedures for a system or component, identify malfunctions, and remedy those malfunctions. Typically described are the diagnostic features of the system or component and the diagnostic tools available for its support. *See also:* **installation manual; operator manual; programmer manual; support manual; user manual.** [610.12]

diagonal microinstruction. A microinstruction capable of specifying a limited number of simultaneous operations needed to carry out a machine language instruction. *Note:* Diagonal microinstructions fall, in size and functionality, between horizontal microinstructions and vertical microinstructions. The designation "diagonal" refers to this compromise rather than to any physical characteristic of the microinstruction. *Contrast with:* **horizontal microinstruction; vertical microinstruction.** [610.12]

dialog (dialogue). Computer-human interaction in which the responses provided by the computer are highly responsive to the questions, answers, and directives given by the user. *See also:* **question-and-answer interaction.** *Syn:* **on-line dialog.** [610.2]

diameter. In image processing, the maximum distance between any two points in a subset of an image. [610.4]

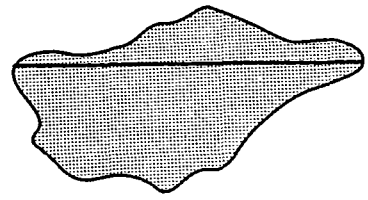


Fig 27
Illustration of Diameter

dibit. Two bits. [610.5]

dichotomizing search. A search in which an ordered set of items is partitioned into two parts, one of which is rejected, and the process is repeated on the accepted part until the search is completed. *See also:* **binary search; Fibonacci search; interpolation search.** [610.5]

dichotomy. A division into two classes that are mutually exclusive and dual in nature. For example, all zero and all nonzero, or all true and all false. [610.1]

dictionary. A list of data items and information about those items, used both to describe and to reference the items. *See also:* **data dictionary; index; table.** *Contrast with:* **directory.** [610.5a]

dictionary/directory. *See:* **data dictionary; data directory.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

DIF. Acronym for **data interchange format.** [610.2]

difference. (1) The result of a subtraction operation. [610.1]
(2) A relational operator that combines two relations having identical attributes and results in a relation containing the tuples that are in the first but not the second relation. *Syn*: **minus; set difference.** *See also*: **intersection; join; product; projection; selection; union.** [610.5]

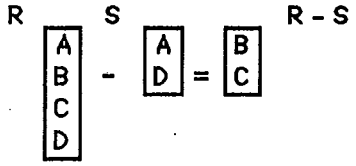


Fig 28
Difference

differential dump. *See:* **change dump.** [610.12]

digit. A symbol or character that represents one of the non-negative integers smaller than the radix; for example, in decimal notation, a digit is one of the characters 0 1 2 3 4 5 6 7 8 9. *Syn*: **numeric character.** [610.1a, 610.5a]

digital. Pertaining to quantities in the form of discrete, integral values. *Contrast with:* **analog.** [610.1]

digital image. An image that has been converted into an array of pixels, each of which has an associated value called its gray level. *Note:* A digital image may be referred to as an image when the intended meaning is clear from the context. *Syn:* **digitized image.** *See also:* **digitization.** [610.4]

digital representation. The representation of numerical quantities by means of digits, or discrete values. *Contrast with:* **analog representation.** [610.1]

digital search tree. A search tree in which the order of the keys is representational of the data contained in the tree. For example, a thumb-index of a dictionary that organizes the alphabet by groups of three or four letters; ABC, DEF, GHI, ... WXYZ. [610.5]

digital simulation. (1) A simulation that is designed to be executed on a digital system. [610.3]
(2) A simulation that is designed to be executed on an analog system but that represents a digital system. [610.3]
(3) A simulation of a digital circuit. *Contrast with:* **analog simulation.** *See also:* **hybrid simulation.** [610.3]

digital sort. A radix sort in which base 10 notation is used. [610.5]

digital tree search. A radix search in which the items in the set to be searched are placed in a tree according to the digital representation of the search keys. *Note*: The tree is traversed in a top-down fashion making comparisons on the bit representations until a match is found for the search argument or the lowest-level node of the tree is encountered. *See also*: **binary tree search.** [610.5]

digital variable.* (1) *See:* **binary variable.** [610.1]
(2) *See:* **integer variable.** [610.1]
*Deprecated

digitization. (1) The process of converting analog data to digital data. [610.1a]
(2) The process of converting an image into a digital image. *See also:* **quantization; sampling.** [610.4]

digitize. To express analog data in digital form. [610.1]

digitized image. *See:* **digital image.** [610.4]

digit place. (1) In a positional notation system, a position corresponding to a given power of the radix. [610.1]
(2) A location in which a digit may occur in a numeral. *Syn:* **digit position; place; position; symbol rank.** [610.1]

digit position. *See:* **digit place.** [610.1]

digit transformation function. A hash function that returns a permutation of the original key with one or more digits removed. For example, in the function below, every other digit is dropped from the original

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

key. [610.5]

Original key	Hash value
964721	942
78394	734

digraph. *See:* **directed graph.** [610.5, 610.12]

diminished-radix complement. The complement obtained by subtracting each digit of a given numeral from the largest digit in the numeration system. For example, ones complement in binary notation, nines complement in decimal notation. *Syn:* **radix-minus one complement; base-minus-one complement; complement on *n*-1.** *Contrast with:* **radix complement.** [610.1]

diminishing increment sort. An insertion sort in which the items in the set to be sorted are divided into subsets, each containing N items; the corresponding items in the subsets are ordered using an insertion sort; and this process is repeated using subsets of diminishing size until the subsets are of size 1. *Syn:* **Shell sort; Shell's method.** [610.5]

direct access. Pertaining to the process of storing and retrieving data using direct access mode. *Contrast with:* **sequential access.** *See also:* **indexed access; indexed sequential access.** [610.5]

direct-access merge sort. An external merge sort in which the auxiliary storage used is direct-access storage. *See also:* **tape merge sort.** [610.5]

direct access method (DAM). An access method in which the access time required to access data is effectively independent of the location of the data. *See also:* **basic direct access method; direct data set.** *Contrast with:* **basic access method; queued access method.** [610.5]

direct access mode. An access mode in which data records are stored and retrieved from storage in such a way that the location of a data record can be derived from the value of some element or elements in the record, regardless of the contents of other data records. *Contrast with:* **indexed sequential access mode; sequential access mode.** [610.5]

direct address. An address that identifies the storage location of an operand. *Syn:* **one-level address.** *Contrast with:* **immediate data; indirect address; *n*-level address.** *See also:* **direct instruction.** [610.12]

direct chaining. *See:* **separate chaining.** [610.5]

direct data set. A file that is accessed using the direct access method. *Contrast with:* **sequential file.** [610.5]

direct insert subroutine. *See:* **open subroutine.** [610.12]

direct instruction. A computer instruction that contains the direct addresses of its operands. *Contrast with:* **immediate instruction; indirect instruction.** *See also:* **absolute instruction; effective instruction.** [610.12]

direct lookup. A table lookup in which the position of an entry is computed as a function of its key value. [610.5]

direct memory access (DMA). Access to data by which data is transferred directly between main memory and storage devices. [610.5]

direct memory access controller (DMAC). The block transfer processor used to implement direct memory access. [610.5]

direct numerical control (DNC). Numerical control in which a dedicated computer controls the operation of the parts programs in a single numerical control machine. [610.2]

directed graph. A graph (2) in which direction is implied in the internode connections. *Syn:* **digraph.** *Contrast with:* **undirected graph.** [610.5a, 610.12a]

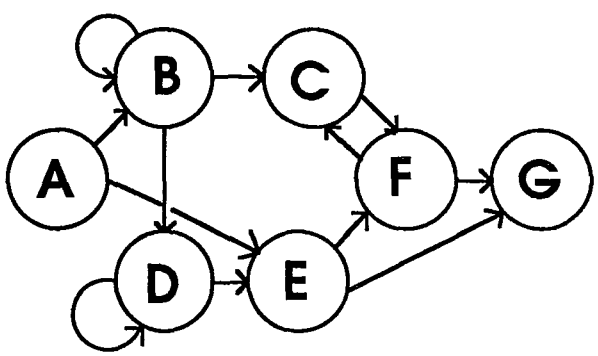


Fig 29
Directed Graph

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

directed rounding. The process of approximating an exact value by a digital numeral such that the resulting error is known to be either non-positive or non-negative. The resulting number is therefore guaranteed to be an upper or lower bound. [610.1]

directory. (1) A list of data items and information about those items, used to reference the items; for example, the directory for each user's personal disk space contains an entry for each file within that space, and a reference to its physical location. *Contrast with:* **dictionary.** *See also:* **data directory; file directory.** [610.5, 610.12a]
(2) *See:* **index.** [610.5]

dirty read. To access data from a storage device or data medium while that same data is being modified by another process. *Syn:* **transient read.** [610.5]

disassemble. To translate an assembled computer program from its machine language version into a form that resembles, but may not be identical to, the original assembly language program. *Contrast with:* **assemble.** [610.12]

disassembler. A software tool that disassembles computer programs. *Syn:* **deassembler.** [610.12]

discrete change model. *See:* **discrete model.** [610.3]

discrete event model. *See:* **discrete model.** [610.3]

discrete model. (1) A mathematical or computational model whose output variables take on only discrete values; that is, in changing from one value to another, they do not take on the intermediate values; for example, a model that predicts an organization's inventory levels based on varying shipments and receipts. *Syn:* **discrete change model; discrete event model; discrete variable model.** [610.3]
(2) A model of a system that behaves in a discrete manner. *Contrast with:* **continuous model.** *See also:* **state machine.** [610.3]

discrete programming. *See:* **integer programming.** [610.2]

discrete simulation. A simulation that uses a discrete model. [610.3]

discrete type. A data type whose members can assume any of a set of distinct values. A discrete type may be an enumeration type or an integer type. [610.12]

discrete variable model. *See:* **discrete model.** [610.3]

discretionary hyphen. In word processing, a hyphen inserted into a word by the user to indicate the desired position for a break, if required by justification. *Note:* If the hyphen is not needed, it does not appear in the formatted text. *Syn:* **ghost hyphen; soft hyphen; syllable hyphen.** *Contrast with:* **required hyphen.** [610.2]

disjunction. (1) The logical 'OR' operator. [610.1a, 610.5]
(2) The result of joining two conditions by the logical 'OR' operator. *Contrast with:* **conjunction.** *See also:* **conjunctive query.** [610.5]

diskette compatibility. The ability of a diskette to be accessed by one or more systems such that data exchange can take place. [610.2]

dispersed data processing (DDP). *See:* **distributed data processing.** [610.2]

display. (1) To present data visually. [610.2]
(2) The result of a display process. *See also:* **copy.** [610.2]

display formatting. A word processing capability that presents the formatted version of a document on a display device. [610.2]

distance teaching. Instruction in which the teacher and student are not in face-to-face contact. Communication is made through correspondence, radio, television, or computer-assisted instruction. [610.2]

distributed database. (1) A database that is not stored in its entirety at a single physical location, but is dispersed over a network of interconnected computers. [610.5]
(2) A database under the overall control of a central database management system, whose storage devices are not all attached to the same processor. [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

distributed data processing (DDP). The use of computers for processing information within a distributed system. *Syn:* **dispersed data processing.** [610.2]

distributed numerical control (DbNC). Numerical control in which a computer controls one or more remote numerical control machines. [610.2]

distributed system. A computer system in which several interconnected computers share the computing tasks assigned to the system. [610.2]

distribution counting sort. An insertion sort in which the sort keys of the items to be sorted fall within some finite range of values, and by counting the number of sort keys having each value, the position of the items in the sorted set can be determined. [610.5]

distribution sort. A sort in which the set to be sorted is divided into two or more subsets such that all items within each subset are within some exclusive range. Each subset is sorted, then the subsets are joined in the correct order. *Syn:* **column sort; distributive sort; pocket sort; separation sort.** *See also:* **digital sort; radix sort.** *Contrast with:* **merge sort.** [610.5]

distributive sort. *See:* **distribution sort.** [610.5]

diverse redundancy. *See:* **diversity.** [610.12]

diversity. In fault tolerance, realization of the same function by different means. For example, use of different processors, storage media, programming languages, algorithms, or development teams. *See also:* **software diversity.** [610.12]

divide-and-conquer sort. *See:* **radix exchange sort.** [610.5]

divide check. An indicator that denotes that an invalid division has been attempted or has occurred. [610.1]

dividend. A number to be divided by another number (the divisor) to produce a result (the quotient), and perhaps a remainder. [610.1]

division transformation function. A hash function that returns the remainder from the division of some value into the original key. For example, in the function below, the original key is divided by 997. [610.5]

Original Key	Calculation	Hash Value
35721	35721/997 = 35 R 826	826
87452	87452/997 = 87 R 713	713

divisor. A number by which another number (the dividend) is divided to produce a result (the quotient), and perhaps a remainder. [610.1]

DMA. Acronym for **direct memory access.** [610.5]

DMAC. Acronym for **direct memory access controller.** [610.5]

DML. Acronym for **data manipulation language.** [610.5]

DNC. Acronym for **direct numerical control.** [610.2]

do-nothing operation. *See:* **no-operation.** [610.12]

document. (1) A medium, and the information recorded on it, that generally has permanence and can be read by a person or a machine. Examples in software engineering include project plans, specifications, test plans, user manuals. *See also:* **document cycle.** [610.2a, 610.12]
(2) To create a document as in (1). [610.12]
(3) To add comments to a computer program. [610.12]

document assembly. In word processing, the assembly of new documents from previously recorded documents or boilerplate text in accordance with specified variables such as names and addresses in iterative documents, or sales figures embedded in the text of a document. *Syn:* **document merge.** [610.2]

document cycle. The steps involved in the creation and handling of a document. *Note:* The cycle typically includes origination, production, reproduction, distribution, fil-

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

ing, and storage. *See also:* **information traffic.** [610.2]

document editor. A text editor used to enter, alter, and view documents. *Syn:* **manuscript editor.** *Contrast with:* **program editor.** [610.2]

document mark. In micrographics, a mark on microfilm, used for counting images or film frames automatically. *Syn:* **blip.**[610.2]

document merge. *See:* **document assembly.** [610.2]

document reference edge. In character recognition, a specified document edge with respect to which the alignment of characters is defined. [610.2]

document traffic. *See:* **information traffic.** [610.2]

documentation. (1) A collection of documents on a given subject. [610.12]
(2) Any written or pictorial information describing, defining, specifying, reporting, or certifying activities, requirements, procedures, or results. [610.12]
(3) The process of generating or revising a document. [610.12]
(4) The management of documents, including identification, acquisition, processing, storage, and dissemination. [610.12]

documentation tree. A diagram that depicts all of the documents for a given system and shows their relationships to one another. *See also:* **specification tree.** [610.12]

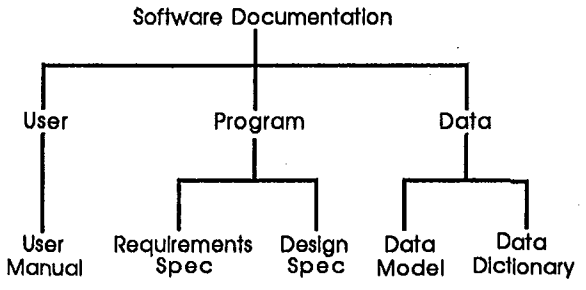


Fig 30
Documentation Tree

domain. (1) The set of all possible values that can be taken on by an independent variable. [610.5]
(2) In a relational data model, the set of all possible values that can be taken on by some attribute. For example, the domain of a three-digit positive integer is [001, 002, 003, ..., 999]. [610.5]

double-ended queue (deque). A list whose contents may be changed by adding or removing items at either end. *Note*: This data structure is inaccurately named because it contradicts the definition of queue. [610.5]

double hashing. Open-address hashing in which collision resolution is handled by using a second hash function to determine a fixed increment and adding multiples of this increment to the original hash value until an empty position is found in the hash table. [610.5]

double length. *See*: **double precision.** [610.1, 610.5]

double-operand instruction. *See:* **two-address instruction.** [610.12]

double precision. Pertaining to the use of two computer words to represent a number in order to preserve or gain precision. *Syn*: **double length.** *Contrast with*: **single precision; multiple precision; triple precision.** [610.1, 610.5]

double-precision addition. Computer addition performed with operands that are expressed in double-precision representation. [610.1]

double-precision arithmetic. Computer arithmetic performed with operands that are expressed in double-precision representation. [610.1]

doublet. A group of two adjacent digits operated upon as a unit. [610.1, 610.5]

doubleword. A contiguous sequence of bits or characters that comprises two computer words and can be addressed as a unit. [610.5]

doubly-chained tree. *See*: **doubly-threaded tree.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

doubly-linked list. A linked list in which each item contains two pointers, one pointing forward to the next item in the list, and one pointing backward to the previous item in the list. *Syn:* **two-way chain.** [610.5]

doubly-threaded tree. A binary tree in which each node contains two link fields; one each for its successor and predecessor nodes with respect to some traversal. *Syn:* **doubly-chained tree.** *Contrast with:* **triply-threaded tree.** [610.5]

down. Pertaining to a system or component that is not operational or has been taken out of service. *Contrast with:* **up.** *See also:* **busy; crash; idle.** [610.12]

download. (1) To transfer some collection of data from a computer memory to another storage location. [610.5]
(2) To transfer some collection of data from the memory of one computer to the memory of a second computer that is relatively smaller than the first; for example, to transfer data from a mainframe computer to a microcomputer. [610.5]

down time. The period of time during which a system or component is not operational or has been taken out of service. *Contrast with:* **up time.** *See also:* **busy time; idle time; mean time to repair; set-up time.** [610.12]

downward compatible. Pertaining to hardware or software that is compatible with an earlier or less complex version of itself; for example, a program that handles files created by an earlier version of itself. *Contrast with:* **upward compatible.** [610.12]

downward compression. In software design, a form of demodularization in which a superordinate module is copied into the body of a subordinate module. *Contrast with:* **lateral compression; upward compression.** [610.12]

DP. Acronym for **data processing.** [610.2]

drift. The latent tendency of control system output to digress from the desired effect. [610.2]

drill and practice interaction. An instruction method employed by some computer-assisted instruction systems, in which the student is asked repeatedly to perform the same or similar tasks. [610.2]

driver. (1) A software module that invokes and, perhaps, controls and monitors the execution of one or more other software modules. *See also:* **test driver.** [610.12]
(2) A computer program that controls a peripheral device and, sometimes, reformats data for transfer to and from the device. [610.12]

DSL. Acronym for **data sublanguage.** [610.5]

DSS. (1) Acronym for **decision support system.** [610.2]
(2) Acronym for **decision support services.** [610.2]

dual coding. *See:* **software diversity.** [610.12]

dual operation. Two Boolean operations are dual if the result of one operation is the negation of the result of applying the other operation to the negated operands, for all combinations of operands. For example, the AND and NOR operations are dual operations. *Contrast with:* **complementary operation.** [610.1]

dummy. Pertaining to a nonfunctioning item used to satisfy some format or logic requirement or to fulfill prescribed conditions. For example, a dummy report containing only titles and column headings with place-holding data instead of real data. [610.5]

dummy data. Data that is used to satisfy some format or logic requirement or to fulfill prescribed conditions. For example, an artificial character used as a placeholder variable within a program. [610.5]

dump. (1) A display of some aspect of a computer program's execution state, usually the contents of internal storage or registers. Types include change dump, dynamic dump, memory dump, postmortem dump, selective dump, snapshot dump; static dump. [610.12]
(2) A display of the contents of a file or device. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

(3) To copy the contents of internal storage to an external medium. [610.12]
(4) To produce a display or copy as in (1), (2), or (3). [610.12]

duodecimal. (1) Pertaining to a selection in which there are 12 possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of 12. [610.1]

duosexadecimal. (1) Pertaining to a selection in which there are 32 possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of 32. *Syn:* **duotricenary.** [610.1]

duotricenary. *See:* **duosexadecimal.** [610.1]

duplicate. To copy data from a source to a destination that has similar physical form as the source. *Syn:* **reproduce.** [610.5]

duplication check. A check that requires that the results of two independent performances of the same operations be identical. [610.5]

dyadic. Pertaining to an operation involving two operands. *Contrast with:* **monadic.** [610.1]

dyadic Boolean operation. A logical operation involving two operands. For example, the equivalence operation. *Contrast with:* **monadic Boolean operation.** [610.1]

dyadic operation. An operation involving two operands. *Contrast with:* **monadic operation.** [610.1]

dyadic operator. An operator that specifies an operation on two operands. *Syn:* **binary operator.** *Contrast with:* **monadic operator.** [610.1]

dyadic selective construct. An if-then-else construct in which processing is specified for both outcomes of the branch. *Contrast with:* **monadic selective construct.** [610.12]

dynamic. Pertaining to an event or process that occurs during computer program execution; for example, dynamic analysis, dynamic binding. *Contrast with:* **static.** [610.12]

dynamic allocation. *See:* **dynamic resource allocation.** [610.12]

dynamic analysis. The process of evaluating a system or component based on its behavior during execution. *Contrast with:* **static analysis.** *See also:* **demonstration; testing.** [610.12]

dynamic binding. Binding performed during the execution of a computer program. *Contrast with:* **static binding.** [610.12]

dynamic breakpoint. A breakpoint whose predefined initiation event is a runtime characteristic of the program, such as the execution of any twenty source statements. *Contrast with:* **static breakpoint.** *See also:* **code breakpoint; data breakpoint; epilog breakpoint; programmable breakpoint; prolog breakpoint.** [610.12]

dynamic buffering. A buffering technique in which the buffer allocated to a computer program varies during program execution, based on current need. *Contrast with:* **simple buffering.** [610.12]

dynamic dump. A dump that is produced during the execution of a computer program. *Contrast with:* **static dump.** *See also:* **change dump; memory dump; postmortem dump; selective dump; snapshot dump.** [610.12]

dynamic error. An error that is dependent on the time-varying nature of an input. *Contrast with:* **static error.** [610.12]

dynamic model. A model of a system in which there is change, such as the occurrence of events over time or the movement of objects through space; for example, a model of a bridge that is subjected to a moving load to determine characteristics of the bridge under changing stress. *Contrast with:* **static model.** [610.3]

dynamic programming. In operations research, a procedure for optimizing a multi-stage problem solution, in which a number of decisions are available at each stage of the process. [610.2]

dynamic relocation. Relocation of a computer program during its execution. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

dynamic resource allocation. A computer resource allocation technique in which the resources assigned to a program vary during program execution, based on current need. [610.12]

dynamic restructuring. The process of restructuring a database, data structure, computer program, or set of system components during program execution. For example, concurrent reorganization of a database. [610.5, 610.12]

dynamic storage allocation. A storage allocation technique in which the storage assigned to a computer program varies during program execution, based on the current needs of the program and of other executing programs. [610.12]

E-R diagram. Acronym for **entity-relationship diagram.** [610.12]

early-failure period. The period of time in the life cycle of a system or component during which hardware failures occur at a decreasing rate as problems are detected and repaired. *Contrast with:* **constant-failure period; wearout-failure period.** *Syn:* **burn-in period.** *See also:* **bathtub curve.** [610.12]

EBCDIC. Acronym for **extended binary-coded decimal interchange code.** [610.1]

EBR. Acronym for **electron beam recording.** [610.2]

echo. (1) To return a transmitted signal to its source, often with a delay to indicate that the signal is a reflection rather than the original. [610.12]
(2) A returned signal, as in (1). [610.12]

echo check. A check in which information that has been transmitted is returned to the information source and compared with the original information to ensure accuracy of the transmission. *Syn:* **read-back check.** [610.5]

ECP. Acronym for **engineering change proposal.** [610.12]

ECR. Acronym for **electronic cash register.** [610.2]

edge. In image processing, a set of pixels belonging to an arc and having the property that pixels on opposite sides of the arc have differing gray levels. *See also:* **border.** [610.4]

edge detection. An image segmentation technique in which edge pixels are identified by examining their neighborhoods. *See also:* **edge linking.** [610.4]

edge enhancement. An image enhancement technique in which edges are sharpened by increasing the contrast between the gray levels of the pixels on opposite sides of the edge. [610.4]

edge image. An image in which each pixel is labeled as either an edge pixel or a non-edge pixel. [610.4]

edge linking. An image processing technique in which neighboring pixels labeled as edge pixels are connected to form an edge. [610.4]

edge operator. A neighborhood operator that determines which pixels in an image are edge pixels. [610.4]

edge pixel. A pixel that lies on an edge. [610.4]

edit. To modify the form or format of computer code, data, or documentation; for example, to insert, rearrange, or delete characters. [610.12]

editing symbol. In micrographics, a symbol on microfilm that is human readable without magnification and that provides cutting, loading, or other preparation instructions. [610.2]

editor. (1) *See:* **text editor.** [610.2, 610.12]
(2) *See:* **linkage editor.** [610.12]

EDP. Acronym for **electronic data processing.** *See:* **automatic data processing.** [610.2]

effective address. The address that results from performing any required indexing, indirect addressing, or other address modification on a specified address. *Note:* If the specified address requires no modification, it is also the effective address. *See also:* **gen-**

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

erated address; indirect address; relative address. [610.12]

effective instruction. The computer instruction that results from performing any required indexing, indirect addressing, or other modification on the addresses in a specified computer instruction. *Note:* If the specified instruction requires no modification, it is also the effective instruction. *See also:* **absolute instruction; direct instruction; immediate instruction; indirect instruction.** [610.12]

efferent. Pertaining to a flow of data or control from a superordinate module to a subordinate module in a software system. *Contrast with:* **afferent.** [610.12]

efficiency. The degree to which a system or component performs its designated functions with minimum consumption of resources. *See also:* **execution efficiency; storage efficiency.** [610.12]

EFTS. Acronym for **electronic funds transfer system.** [610.2]

egoless programming. A software development technique based on the concept of team, rather than individual, responsibility for program development. Its purpose is to prevent individual programmers from identifying so closely with their work that objective evaluation is impaired. [610.12]

EIS. Acronym for **executive information system.** *See:* **management information system.** [610.2]

EITHER-OR.* *See:* **OR.** [610.1]
*Deprecated.

electron beam recording (EBR). In micrographics, a specific method of producing computer output microfilm in which a beam of electrons is directed onto an energy-sensitive microfilm. [610.2]

electronic bulletin board. In an electronic mail system, a storage area shared by several users, each having access to all messages left in that area. [610.2]

electronic cash register (ECR). A device that functions as both a cash register and a point-of-sale terminal to a central computer performing inventory control, price updating, and other retail sales functions. [610.2]

electronic data processing (EDP). *See:* **automatic data processing.** [610.2]

electronic funds transfer system (EFTS). A data collection and telecommunication system that electronically transports information about the movement of funds between accounts managed by financial institutions. [610.2]

electronic mail. (1) The generation, transmission, and display of correspondence and documents by electronic means. *See also:* **electronic bulletin board; electronic mailbox.** *Syn:* **mailbox service.** [610.2]
(2) The concepts and technologies employed for the electronic communication of textual material. [610.2]

electronic mailbox. A storage area used to hold all messages addressed to a particular user of an electronic mail system. [610.2]

electronic office. An office that makes use of office automation. *Syn:* **automated office; office of the future.** *See also:* **paperless office.** [610.2]

electronic spread sheet. (1) A computer program that enables the user to set up a display of rows and columns in which some entries are manually entered and others are calculated automatically using formulas supplied by the user. *Syn:* **spread sheet.** [610.2]
(2) The display of rows and columns produced by a computer program as in (1). [610.2]

element. *See*: **binary element; data element.** [610.5]

embedded computer system. A computer system that is part of a larger system and performs some of the requirements of that system; for example, a computer system used in an aircraft or rapid transit system. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

embedded data dictionary. *See*: **active data dictionary.** [610.5]

embedded hyphen. *See:* **required hyphen.** [610.2]

embedded software. Software that is part of a larger system and performs some of the requirements of that system; for example, software used in an aircraft or rapid transit system. [610.12]

empirical. Pertaining to information that is derived from observation, experiment, or experience. [610.3]

empty medium. A data medium that does not contain data. [610.5]

emulate. To represent a system by a model that accepts the same inputs and produces the same outputs as the system represented. For example, to emulate an 8-bit computer with a 32-bit computer. *See also:* **simulate.** [610.3]

emulation. (1) A model that accepts the same inputs and produces the same outputs as a given system. *See also:* **simulation.** [610.3, 610.12]
(2) The process of developing or using a model as in (1). [610.3, 610.12]

emulator. A device, computer program, or system that performs emulation. [610.3, 610.12a]

encapsulation. A software development technique that consists of isolating a system function or a set of data and operations on those data within a module and providing precise specifications for the module. *See also:* **data abstraction; information hiding.** [610.12]

encode. To represent data in symbolic form using a code or a coded character set such that reconversion to the original form is possible. *Note*: Sometimes used when complete reconversion is not possible. *Contrast with*: **decode.** *See also*: **code.** [610.5]

encode-decode table. *See*: **code-decode table.** [610.5]

encoded data. *See*: **code (4).** [610.5]

encoder. A device or system that encodes data. *Contrast with*: **decoder.** [610.5]

end-around carry. A carry process in which a carry digit generated in the most significant digit place is added directly to the least significant digit place. For example, when adding two negative numbers using nines complement. [610.1]

end-around carry shift. *See:* **circular shift.** [610.1a]

end-around shift. *See:* **circular shift.** [610.1]

end of file (EOF). An internal label, immediately following the last record of a file, signalling the end of that file. *Syn*: **end-of-file.** [610.5]

end user. *See:* **user.** [610.2]

end user computing. The performance of system development and data processing tasks by the user of a computer system. *Syn:* **user-driven computing.** [610.2]

end-of-file. *See*: **end of file.** [610.5]

end-of-volume label. An internal label that precedes and initiates the beginning of the data contained in that volume. *Syn*: **volume label.** [610.5]

end-point. *See*: **terminal node.** [610.5]

end-user language. *See*: **query language.** [610.5]

endogenous variable. A variable whose value is determined by conditions and events within a given model. *Syn:* **internal variable.** *Contrast with:* **exogenous variable.** [610.3]

endorder traversal. *See*: **postorder traversal.** [610.5]

engineering. The application of a systematic, disciplined, quantifiable approach to structures, machines, products, systems, or processes. [610.12]

engineering change. In configuration management, an alteration in the configuration of a configuration item or other designated item after formal establishment of its con-

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

figuration identification. *See also:* **configuration control; engineering change proposal.** *Contrast with:* **deviation; waiver.** [610.12]

engineering change proposal (ECP). In configuration management, a proposed engineering change and the documentation by which the change is described and suggested. *See also:* **configuration control.** [610.12]

enhancement. *See:* **image enhancement.** [610.4]

enqueue. To append an item to a queue. *Contrast with:* **dequeue.** [610.5]

enterprise view. *See:* **conceptual view.** [610.5]

entity. (1) In computer programming, any item that can be named or denoted in a program. For example, a data item, program statement, or subprogram. [610.12]
(2) In data management, a distinguishable object, either real or abstract, about which data are recorded; for example, a person such as a CUSTOMER, or a concept, such as SALES-REVENUE, about which data is stored in a data structure. *Syn:* **entity instance.** [610.5]

entity attribute. A named characteristic or property of a design entity. It provides a statement of fact about the entity. [610.12]

entity/attribute matrix. A representation of a relation in the form of a matrix such that each row represents an entity and each column represents an attribute of the entity. (See Fig 11, Entity/Attribute Matrix *Students.*) [610.5]

attributes → R

No.	Name	Grade	Homeroom
15	Mary	4	26A
20	Joe	6	43
21	Harry	4	27
27	Michael	5	25
30	Susan	5	25
42	Mickey	6	41

entity ↓

Fig 31
Entity/Attribute Matrix *Students*

entity class. *See:* **entity set.** [610.5]

entity instance. *See:* **entity.** [610.5]

entity-relationship data model. A logical view of data within a system, representing the entities in the system as well as relationships among the entities, attributes of the entities, and attributes of the relationships. [610.5]

entity-relationship (E-R) diagram. A diagram that depicts a set of real-world entities and the logical relationships among them. *Syn:* **entity-relationship map.** *See also:* **data structure diagram.** [610.12]

entity-relationship (E-R) map. *See:* **entity-relationship diagram.** [610.12]

entity set. A collection of entities that have similar properties, such as a set of CUSTOMERS. *Syn:* **entity class; entity type.** [610.5]

entity type. *See:* **entity set.** [610.5]

entrance. *See:* **entry point.** [610.12]

entry. (1) An element of information in a data structure, that describes an identifiable entity; for example, a member of a table, list, or queue. *See also:* **data entry.** [610.5]
(2) *See:* **entry point.** [610.12]

entry point. A point in a software module at which execution of the module can begin. *Contrast with:* **exit.** *Syn:* **entrance; entry.** *See also:* **reentry point.** [610.12]

enumeration type. A discrete data type whose members can assume values that are explicitly defined by the programmer. For example, a data type called COLORS with possible values RED, BLUE, and YELLOW. *Contrast with:* **character type; integer type; logical type; real type.** [610.12]

environment. The external objects, conditions, and processes that influence the behavior of a system. [610.3]

environmental simulation. A simulation that depicts all or part of the natural or man-

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

made environment of a system; for example, a simulation of the radar equipment and other tracking devices that provide input to an aircraft tracking system. [610.3]

EOF. Acronym for **end of file.** [610.5]

epilog breakpoint. A breakpoint that is initiated upon exit from a given program or routine. *Syn:* **postamble breakpoint.** *Contrast with:* **prolog breakpoint.** *See also:* **code breakpoint; data breakpoint; dynamic breakpoint; programmable breakpoint; static breakpoint.** [610.12]

equal interval quantizing. A quantization technique in which the range of gray levels in an image is divided into intervals of equal length and the quantization level assigned to each pixel is the same for all pixels whose original gray levels fall within the same interval. *Syn:* **linear quantizing.** [610.4]

equal probability quantizing. A quantization technique in which the range of gray levels in an image is divided into contiguous intervals such that the frequency of occurrence of each quantization level is the same. *Syn:* **histogram equalization.** [610.4]

equilibrium. *See:* **steady state.** [610.3]

equilibrium condition. *See:* **final condition.** [610.3]

equivalence. A dyadic Boolean operator having the property that if P is a statement and Q is a statement, then the equivalence of P and Q is true if and only if both statements are true or both statements are false. *Note:* The equivalence of P and Q is often represented by $P \equiv Q$. *Syn:* **exclusive NOR; IF-AND-ONLY-IF.** [610.1]

P	Q	P≡Q
0	0	1
0	1	0
1	0	0
1	1	1

Fig 32
Equivalence Truth Table

equivalent binary digit(s). The number of binary digits required to represent a number expressed in another numeration system with no loss of precision. *Note:* This number is approximately 31/3 times the number of decimal digits. *Syn:* **equivalent binary digit factor.** [610.1]

equivalent binary digit factor. *See:* **equivalent binary digits.** [610.1]

equivalent faults. Two or more faults that result in the same failure mode. [610.12]

erase. *See:* **delete.** [610.5]

erase character. * *See:* **delete character.** * Deprecated. [610.5]

error. (1) The difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. For example, a difference of 30 meters between a computed result and the correct result. [610.12]
(2) An incorrect step, process, or data definition. For example, an incorrect instruction in a computer program. [610.12]
(3) An incorrect result. For example, a computed result of 12 when the correct result is 10. [610.12]
(4) A human action that produces an incorrect result. For example, an incorrect action on the part of a programmer or operator. [610.12]
Note: While all four definitions are commonly used, one distinction assigns definition 1 to the word "error," definition 2 to the word "fault," definition 3 to the word "failure," and definition 4 to the word "mistake." *See also:* **dynamic error; fatal error; indigenous error; semantic error; syntactic error; static error; transient error.** [610.12]

error burst. A group of bits in which two erroneous bits are separated by fewer than a specified number of correct bits. [610.1]

error character. A control character used to indicate that an error exists in the data or has occurred during transmission. [610.5]

error control character. *See:* **accuracy control character.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

error-correcting code. A code containing redundant information that can be used to detect certain classes of errors and to restore a word, byte, character, quantity, or message to its correct representation. *Syn:* **error-detecting and correcting code.** [610.1]

error-detecting and correcting code. *See:* **error-correcting code.** [610.1]

error-detecting code. A code containing redundant information that can be used to detect certain classes of errors in a word, byte, character, quantity, or message. *Syn:* **self-checking code.** [610.1]

error model. (1) A model used to estimate or predict the extent of deviation of the behavior of an actual system from the desired behavior of the system; for example, a model of a communications channel, used to estimate the number of transmission errors that can be expected in the channel. [610.3]
(2) In software evaluation, a model used to estimate or predict the number of remaining faults, required test time, and similar characteristics of a system. *Syn:* **error prediction model.** [610.3, 610.12]

error prediction model. *See:* **error model.** [610.3, 610.12]

error prediction. A quantitative statement about the expected number or nature of faults in a system or component. *See also:* **error model; error seeding.** [610.12]

error seeding. The process of intentionally adding known faults to those already in a computer program for the purpose of monitoring the rate of detection and removal, and estimating the number of faults remaining in the program. *Syn:* **bug seeding; fault seeding.** *See also:* **indigenous error.** [610.12]

error tolerance. The ability of a system or component to continue normal operation despite the presence of erroneous inputs. *See also:* **fault tolerance; robustness.** [610.12]

ESC. Abbreviation for the **escape character.** [610.5]

escape character (ESC). A code extension character used, in some cases with one or more succeeding characters, to indicate by some convention or agreement that the coded representations following the character or the group of characters are to be interpreted according to a different code or according to a different coded character set. [610.5]

estimated entry search. *See:* **interpolation search.** [610.5]

even-odd check. *See:* **parity check.** [610.1]

even parity. (1) An error detection method in which the number of ones in a binary word, byte, character, or message is maintained as an even number. [610.1]
(2) The property possessed by a binary word, byte, character, or message that has an even number of ones. [610.1]

event. (1) An occurrence that causes a change of state in a simulation. *See also:* **conditional event; time-dependent event.** [610.3]
(2) The instant in time at which a change in some variable occurs. [610.3]

event-driven simulation. *See:* **event-oriented simulation.** [610.3]

event-oriented simulation. A simulation in which attention is focused on the occurrence of events and the times at which those events occur; for example, a simulation of a digital circuit that focuses on the time of state transition. *Syn:* **event-driven simulation; event-sequenced simulation.** [610.3]

event-sequenced simulation. *See:* **event-oriented simulation.** [610.3]

exception. An event that causes suspension of normal program execution. Types include addressing exception, data exception, operation exception, overflow exception, protection exception, underflow exception. [610.12]

exception reporting. An information processing technique that screens large amounts of computerized data and produces a report containing only the data that require action. *See also:* **information overload.** [610.2]

except operation.* *See:* **exclusion.** [610.1]
*Deprecated.

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

excess-fifty code. A binary code in which a decimal number n is represented by the binary equivalent of $n + 50$. *Syn:* **excess-fifty representation.** [610.1]

excess-fifty representation. *See:* **excess-fifty code.** [610.1]

excess-sixty-four code. A binary code in which a decimal number n is represented by the binary equivalent of $n + 64$. *Syn:* **excess-sixty-four representation.** [610.1]

excess-sixty-four representation. *See:* **excess-sixty-four code.** [610.1]

excess-three BCD. *See:* **excess-three code.** [610.1]

excess-three code. A BCD code in which a decimal digit n is represented by the four-bit binary equivalent of $n + 3$. *Syn:* **excess-three BCD; excess-three representation.** [610.1]

DECIMAL DIGIT:	0	1	2	3
EXCESS 3 CODE:	0011	0100	0101	0110

Fig 33
Excess-Three Code

excess-three representation. *See:* **excess-three code.** [610.1]

exchange data. Data that is received or transmitted via data exchange in an appropriate format. *See also*: **data exchange.** [610.5]

exchange selection sort. *See*: **bubble sort.**[610.5]

exchange sort. A sort in which pairs of items in a set are examined in some sequence, pairs found out of order are exchanged, and the process is repeated until all items are in the correct order. Multiple passes are usually required. *See also*: **Batcher's parallel sort; bubble sort; cocktail shaker sort; quicksort; radix exchange sort.** [610.5]

exclusion. A dyadic Boolean operator having the property that if P is a statement and Q is a statement, then the expression P exclusion Q is true if and only if P is true and Q is false. *Note:* P exclusion Q is often represented by a combination of AND and NOT symbols such as $P \wedge \sim Q$. *Syn:* **NOT-IF-THEN.**[610.1]

P	Q	$P \wedge \sim Q$
0	0	0
0	1	0
1	0	1
1	1	0

Fig 34
Exclusion Truth Table

exclusive lock. A lock that grants the holder sole access to the locked data. No other process can access the data for either read or write purposes. *Contrast with*: **shared lock.** [610.5]

exclusive NOR (XNOR). *See:* **equivalence.** [610.1]

exclusive OR (XOR). A dyadic Boolean operator having the property that if P is a statement and Q is a statement, then P exclusive-OR Q is true if and only if either, but not both, is true. *Note:* P exclusive OR Q is often represented by $P \oplus Q$ or $P \veebar Q$. *Syn:* **nonequivalence; inequivalence; modulo-two sum.** *Contrast with:* **OR.** [610.1a]

P	Q	$P \oplus Q$
0	0	0
0	1	1
1	0	1
1	1	0

Fig 35
Exclusive OR Truth Table

execute. To carry out an instruction, process, or computer program. [610.12]

execution efficiency. The degree to which a system or component performs its designated functions with minimum consumption of time. *See also:* **execution time; storage efficiency.** [610.12]

execution monitor. *See:* **monitor.** [610.12]

execution time. The amount of elapsed time or processor time used in executing a computer program. *Note:* Processor time is usually

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052824 9

less than elapsed time because the processor may be idle (for example, awaiting needed computer resources) or employed on other tasks during the execution of a program. *Syn:* **run time (3); running time.** *See also:* **overhead time.** [610.12]

execution trace. A record of the sequence of instructions executed during the execution of a computer program. Often takes the form of a list of code labels encountered as the program executes. *Syn:* **code trace; control-flow trace.** *See also:* **retrospective trace; subroutine trace; symbolic trace; variable trace.** [610.12]

executive. *See:* **supervisory program.** [610.12]

executive information system (EIS). *See:* **management information system.** [610.2]

executive program. *See:* **supervisory program.** [610.12]

executive state. *See:* **supervisor state.** [610.12]

exit. A point in a software module at which execution of the module can terminate. *Contrast with:* **entry point.** *See also:* **return.** [610.12]

exit routine. A routine that receives control when a specified event, such as an error, occurs. [610.12]

exogenous variable. A variable whose value is determined by conditions and events external to a given model. *Syn:* **external variable.** *Contrast with:* **endogenous variable.** [610.3]

expandability. *See:* **extendability.** [610.12]

explicit address. *See:* **absolute address.** [610.12]

exponent. (1) A superscript indicating the number of times a number is to be used as a factor. [610.1]
(2) The component of a floating-point number that normally signifies the integer power to which the radix is raised in determining the value of the represented number. *Syn:* **characteristic; exrad; floating-point coefficient.** *Contrast with:* **significand.** [610.1]

exponent character. (1) A character within a picture specification that represents the beginning of the exponent within a floating point number. *Note*: K and E are commonly used. [610.5]
(2) A character within a picture specification that represents the scaling factor for a decimal number. Specified with an integer constant, it indicates the number of decimal positions the decimal point is to be moved from its assumed position to the right (if the constant is positive) or to the left (if the constant is negative). *Note*: F is commonly used. [610.5]

exponent overflow. A condition that occurs in floating-point arithmetic if an attempt is made to create an exponent greater than the largest positive number that can be processed or stored. *Syn:* **characteristic overflow.** [610.1]

exponent spill. A condition that occurs in floating-point arithmetic when the exponent of a computed result lies outside the range that can be processed or stored. [610.1]

exponent underflow. A condition that occurs in floating-point arithmetic if an attempt is made to create a negative exponent greater in absolute value than the smallest nonzero number that can be processed or stored. *Syn:* **characteristic underflow.** [610.1]

expression. A sequence of constants, variables, and functions connected by operators to indicate a desired computation. [610.1]

exrad. *See:* **exponent (2).** [610.1]

extendability. The ease with which a system or component can be modified to increase its storage or functional capacity. *Syn:* **expandability; extensibility.** *See also:* **flexibility; maintainability.** [610.12]

extended binary coded decimal interchange code (EBCDIC). A binary code in which 256 letters, numbers, and special characters are represented by eight-bit numerals. [610.1]

extended binary tree. A full binary tree in which all terminal nodes contain data. [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

extended precision. *See:* **multiple precision.** [610.1]

extensibility. *See:* **extendability.** [610.12]

extent. A continuous area of storage on a direct access data medium, occupied by or reserved for a particular file. *See also*: **primary space allocation; secondary space allocation.** [610.5]

external chaining. *See*: **separate chaining.** [610.5]

external data file. (1) A data file that is sorted on an external storage medium such as a magnetic tape. [610.5]
(2) A data file that is stored apart from the system using the data. [610.5]

external data model. A data model depicting entities within a specific application or type of application in an organization. *Contrast with*: **internal model.** [610.5]

external data submodel. *See*: **external schema.** [610.5]

external entry search. *See*: **interpolation search.** [610.5]

external label. A label, usually not machine-readable, attached to a data medium container; for example, a paper sticker attached to the outside of a reel of magnetic tape. *Contrast with*: **internal label.** [610.5]

external merge sort. A merge sort that makes use of auxiliary storage. *See also*: **balanced merge sort; direct-access merge sort; multiway merge sort; oscillating sort; tape merge sort; unbalanced merge sort.** *Contrast with*: **internal merge sort.** [610.5]

external node. *See*: **terminal node.** [610.5]

external record. A record within an external view. [610.5]

external schema. (1) A description of the format, layout, and contents of the data, within a database, to be employed by a user or application program. *Note*: The schema is written using the data definition portion of the data sublanguage. *Syn*: **logical view; data submodel.** *See also*: **external database model.** *Contrast with*: **conceptual schema; internal schema.** [610.5]
(2) A logical description of an organization or enterprise. *Note*: The external schema may differ from the conceptual schema in that some entities, attributes, or relationships may be omitted, renamed, or otherwise transformed. [610.5]
(3) A description of the user's view of data. *Syn*: **external data submodel; view; subschema.** [610.5]

external sort. A sort that requires the use of auxiliary storage. *Contrast with*: **internal sort.** [610.5]

external variable. *See:* **exogenous variable.** [610.3]

external view. The format, layout, and contents of the data in a database that a user or application program uses, as described in an external schema. *See also*: **external record.** *Contrast with*: **conceptual schema.** [610.5]

extract. To pick, from a set of items, all items that meet a particular criterion. *See also*: **database extract; select.** [610.5]

extraction indexing. *See:* **derivative indexing.** [610.2]

F format. Abbreviation for **fixed format.** [610.5]

fac. Abbreviation for **facsimile.** [610.2a]

facsimile (fac or fax). (1) An exact copy or likeness. [610.2]
(2) The process by which fixed graphic images are scanned, transmitted electronically, and reproduced either locally or remotely. [610.2]
(3) The result of the process in (2). *See also:* **microfacsimile.** [610.2]

facsimile telegraphy. A facsimile transmission system designed specifically for the transmission of photographic images. The reproduction may be in two significant states only (for example, black and white), may contain intermediate shades, or may be colored. *Syn:* **telephotography.** [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052826 2

facsimile terminal. A terminal used in facsimile transmission. [610.2]

facsimile transmission. The use of a telecommunication system to transmit fixed graphic images. *See also:* **facsimile, facsimile telegraphy.** *Syn:* **telefax.** [610.2]

factor. (1) Any of the operands in a multiplication operation. [610.1]
(2) A number used as a multiplier to cause a set of quantities to fall within a given range of values. *Syn:* **factor scale.** [610.1]

factoring. (1) The process of decomposing a system into a hierarchy of modules. *See also:* **modular decomposition.** [610.12]
(2) The process of removing a function from a module and placing it into a module of its own. [610.12]

factor scale. *See:* **factor (2).** [610.1]

fail safe. Pertaining to a system or component that automatically places itself in a safe operating mode in the event of a failure; for example, a traffic light that reverts to blinking red in all directions when normal operation fails. *Contrast with:* **fail soft.** *See also:* **fault secure; fault tolerance.** [610.12]

fail soft. Pertaining to a system or component that continues to provide partial operational capability in the event of certain failures; for example, a traffic light that continues to alternate between red and green if the yellow light fails. *Contrast with:* **fail safe.** *See also:* **fault secure; fault tolerance.** [610.12]

failure. The inability of a system or component to perform its required functions within specified performance requirements. *Note:* The fault tolerance discipline distinguishes between a human action (a mistake), its manifestation (a hardware or software fault), the result of the fault (a failure), and the amount by which the result is incorrect (the error). *See also:* **crash; dependent failure; exception; failure mode; failure rate; hard failure; incipient failure; independent failure; random failure; soft failure; stuck failure.** [610.12]

failure mode. The physical or functional manifestation of a failure. For example, a system in failure mode may be characterized by slow operation, incorrect outputs, or complete termination of execution. [610.12]

failure rate. The ratio of the number of failures of a given category to a given unit of measure; for example, failures per unit of time, failures per number of transactions, failures per number of computer runs. *Syn:* **failure ratio.** [610.12]

failure ratio. *See:* **failure rate.** [610.12]

false add. *See:* **OR.** [610.1]

false alarm. *See:* **false identification.** [610.4]

false identification. In pattern classification, the assignment of a pattern to a pattern class other than its true pattern class. *Syn:* **false alarm; type II error.** *Contrast with:* **misidentification.** [610.4]

fast time. (1) Simulated time with the property that a given period of actual time represents more than that period of time in the system being modeled; for example, in a simulation of plant growth, running the simulation for one second may result in the model advancing time by one full day; that is, simulated time advances faster than actual time. [610.3]
(2) The duration of activities within a simulation in which simulated time advances faster than actual time. *Contrast with:* **real time; slow time.** [610.3]

fatal error. An error that results in the complete inability of a system or component to function. [610.12]

father. *See:* **parent node.** [610.5]

father file. A file that contains data that have since been updated in another file, called the son file. *See also:* **grandfather file.** [610.5]

fault. (1) A defect in a hardware device or component; for example, a short circuit or broken wire. [610.12]
(2) An incorrect step, process, or data definition in a computer program. *Note:* This definition is used primarily by the fault

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

tolerance discipline. In common usage, the terms "error" and "bug" are used to express this meaning. *See also:* **data-sensitive fault; program sensitive fault; equivalent faults; fault masking; intermittent fault.** [610.12]

fault dictionary. A list of faults in a system or component, and the tests that have been designed to detect them. [610.12]

fault masking. A condition in which one fault prevents the detection of another. [610.12]

fault secure. Pertaining to a system or component in which no failures are produced from a prescribed set of faults. *See also:* **fault tolerance; fail safe; fail soft.** [610.12]

fault seeding. *See:* **error seeding.** [610.12]

fault tolerance. (1) The ability of a system or component to continue normal operation despite the presence of hardware or software faults. *See also:* **error tolerance; fail safe; fail soft; fault secure; robustness.** [610.12]
(2) The number of faults a system or component can withstand before normal operation is impaired. [610.12]
(3) Pertaining to the study of errors, faults, and failures, and of methods for enabling systems to continue normal operation in the presence of faults. *See also:* **recovery (2); redundancy; restart.** [610.12a]

fault tolerant. Pertaining to a system or component that is able to continue normal operation despite the presence of faults. [610.12]

fax. Abbreviation for **facsimile.** [610.2a]

FCA. Acronym for **functional configuration audit.** [610.12]

FCFS. Acronym for **first-come, first-served.** *See:* **first-in, first-out.** [610.5]

FE. Acronym for **format effector character.** [610.5]

feasibility. The degree to which the requirements, design, or plans for a system or component can be implemented under existing constraints. [610.12]

feature. (1) In pattern recognition, an attribute of a pattern that may contribute to pattern classification; for example, size, texture, or shape. [610.4]
(2) *See:* **software feature.** [610.128]

feature extraction. A step in pattern recognition, in which measurements or observations are processed to find attributes that can be used to assign patterns to pattern classes. [610.4]

feature space. In pattern recognition, a set of all possible n-tuples $(x_1, x_2, ..., x_n)$ that can be used to represent n features of a pattern. *See also:* **measurement space.** [610.4]

feedback. That portion of the output of a control system used as input for another phase of the system, particularly for self-correcting, self-regulating, or control purposes, as in closed-loop control. [610.2]

feedback control. *See:* **closed-loop control.** [610.2]

FEFO. Acronym for **first-ended, first-out.** [610.5]

fetch. To locate and load computer instructions or data from storage. *See also:* **move; store.** [610.12]

FF. Acronym for **form feed character.** [610.5]

Fibonacci number. An integer in the Fibonacci series. [610.1]

Fibonacci search. A dichotomizing search in which, at each step in the search, the set of items is partitioned in accordance with the Fibonacci series. For example, a set of 8 items is partitioned to 5 and 3, the subset of 5 is partitioned to 3 and 2, and so on. If the number of items in the original set is other than a Fibonacci number, the next higher Fibonacci number is used to partition the set. *Contrast with:* **binary search; interpolation search.** [610.5]

Fibonacci series. A series of integers formulated by the Italian mathematician Leonardo Fibonacci, in which each integer is equal to the sum of the two preceding

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052828 6

integers in the series, that is, 0, 1, 1, 2, 3, 5, 8, 13,.... Represented mathematically by

$$x_i = x_{i-1} + x_2$$

where

$x_0 = 0$

$x_1 = 1$ [610.1]

fiche. *See:* **microfiche.** [610.2]

fidelity. The degree of similarity between a model and the system properties being modeled. *Syn:* **correspondence.** *See also:* **model validation.** [610.3]

field. (1) A specified area within a record, used for a particular data item; for example a group of card columns in which a telephone number is recorded. [610.5]
(2) The smallest unit of data that can be referred to in a database. *See also*: **database segment.** [610.5]

field length. The number of words or characters in a field. [610.5]

field length type. An indication of whether the field is fixed or variable in length. *Note*: If a field is a variable length type, the field length expresses the maximum length possible. [610.5]

field-locking. *See*: **lock.** [610.5]

field separator. A character or byte used to identify a boundary between two fields. [610.5]

FIFO. Acronym for **first-in, first-out.** [610.5]

fifth generation language (5GL). A computer language that incorporates the concepts of knowledge-based systems, expert systems, inference engines, and natural language processing. *Contrast with:* **assembly language; fourth generation language; high-order language; machine language.** *Note:* Specific languages will be defined in P610.13. [610.12, P610.13]

fifth normal form (5NF). One of the forms used to characterize relations; a relation is said to be in fifth normal form if it is in fourth normal form and if every join dependency in the relation is a consequence only of the candidate keys of the relation. *Syn*: **projection/join normal form.** [610.5]

figurative constant. A data name that is reserved for a specific constant in a programming language. For example, the data name THREE may be reserved to represent the value 3. *See also:* **literal.** [610.12]

file. A set of related records treated as a unit. For example, in stock control, a file could consist of a set of invoice records. *See also*: **data file; data set; logical file.** [610.5, 610.12]

file access mode. The type of access allowed for a given file and a given user. For example, the file access mode for a given file might be read-only access for one user, and read/write access for another. *Syn*: **access type.** [610.5]

file attribute. A property, feature, or characteristic of a file. [610.5]

file cleanup. The removal of superfluous data from a file. *Syn*: **file tidying.** [610.5]

file directory. (1) A list of files and their locations within a computer system. *See also*: **catalog.** [610.5]
(2) A list of the files and their locations on a particular storage device or volume. [610.5]

file gap. An unused area on a data medium between the end of one file or group of data and the beginning of another file or group of data. [610.5]

file layout. The arrangement and structure of data in a file. *Syn*: **file organization.** [610.5]

file-locking. *See*: **lock.** [610.5]

file maintenance. The activity of adding, changing, or deleting data in a file as needed. [610.5]

file name. (1) One or more characters used to identify a file. [610.5]
(2) A name associated with a set of file data or output data. [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

file organization. (1) *See*: **file layout.** [610.5] (2) The order of physical records within a file that determines the access method to be implemented in order to use the file. [610.5]

file processing. The periodic updating of one or more master files to reflect the effects of current data, often from a transaction file. For example, a monthly run updating the master inventory file. [610.2]

file tidying. *See*: **file cleanup.** [610.5]

fill. *See*: **character fill; filler character; zero fill.** [610.5]

filler. One or more data items adjacent to an item of data that forces that item to take on a specified size; for example, in an 80-character output record in which a 30-character NAME, 20-character ADDRESS, and a 3-character AGE is to be placed, filler would be used to expand the data to be 80 characters. *See also*: **character fill; pad; padding.** [610.5]

filler character. (1) A character used to occupy an area on a printed medium; for example, on a legal document, dashes or asterisks used to fill out a field to ensure that nothing is added to the field once the document has been issued. *See also*: **filler.** [610.5] (2) A character that does not itself convey data but that may delete unwanted data, as in blanks used to fill out a field. *See also*: **character fill.** [610.5]

film frame. In micrographics, a line on microfilm, perpendicular to the document reference edge, on which binary characters may be written or read. [610.2]

FILO. Acronym for **first-in, last-out.** *See*: **last-in, first-out.** [610.5]

final condition. The values assumed by the variables in a system, model, or simulation at the completion of some specified duration of time. *Syn:* **equilibrium condition.** *Contrast with:* **boundary condition; initial condition.** [610.3]

final state. The values assumed by the state variables of a system, component, or simulation at the completion of some specified duration of time. *Contrast with:* **initial state.** [610.3]

finite state machine. A computational model consisting of a finite number of states and transitions between those states, possibly with accompanying actions. [610.12]

firmware. The combination of a hardware device and computer instructions and data that reside as read-only software on that device. *Notes:* (1) This term is sometimes used to refer only to the hardware device or only to the computer instructions or data, but these meanings are deprecated. [610.12] (2) The confusion surrounding this term has led some to suggest that it be avoided altogether. [610.12]

first-come, first-served (FCFS). *See*: **first-in, first-out (FIFO).** [610.5]

first-ended, first-out (FEFO). A queueing technique for concurrent processes in which items are retrieved from the queue based on the time at which the item is placed completely in the queue. That is, the item whose final segment is placed in the queue before those of all other items, will exit the queue before those other items. *Note*: Often used in message queueing applications. [610.5]

first generation language (1GL). *See:* **machine language.** [610.12]

first-in, first-out (FIFO). (1) A technique for managing a set of items to which additions and deletions are to be made; items are appended to one end of a list and retrieved from the other end. *See also*: queue. [610.5] (2) Pertaining to a system in which the next item to exit the system is the item that has been in the system for the longest time. *Contrast with*: **last-in, first-out.** *Syn*: **first-come, first-served.** [610.5]

first-in, last-out. *See*: **last-in, first-out.** [610.5]

first normal form (1NF). One of the forms used to characterize relations; a data struc-

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052830 4

ture or relation is said to be in first normal form if it has no repeating groups. For example:

UNNORMALIZED

ORDER0 = {ORDER-NO} + DATE + CUSTOMER-NO
+ CUSTOMER-NAME + CUSTOMER-ADDRESS
+ ((SEQUENCE-NO + ITEM-NO + ITEM-DESCRIPTION
+ QUANTITY-ORDERED + UNIT-PRICE
+ EXTENDED-PRICE)) + TOTAL-ORDER-AMOUNT

FIRST NORMAL FORM

ORDER1 = {ORDER-NO} + DATE + CUSTOMER-NO
+ CUSTOMER-NAME + CUSTOMER-ADDRESS
+ TOTAL-ORDER-AMOUNT

ITEM1 = {ORDER-NO + SEQUENCE-NO} + ITEM-NO
+ ITEM-DESCRIPTION + QUANTITY-ORDERED
+ UNIT-PRICE + EXTENDED-PRICE

Note: repeating group enclosed in parenthesis. Keys in brackets.

Fig 36
First Normal Form [610.5]

five-bit byte. *See*: **quintet.** [610.5]

fix. To convert a number from floating-point representation to fixed-point representation. *Contrast with:* **float.** [610.1]

fixed. *See*: **read-only access.** [610.5]

fixed binary data. *See*: **fixed-point binary data.** [610.5]

fixed decimal data. *See*: **fixed-point real data.** [610.5]

fixed format (F format) . A file organization in which all logical records in the file are of fixed length. *Contrast with*: **variable format.** [610.5]

fixed length. Pertaining to a record or field that has a constant length, regardless of the specific data contained in it. Filler characters may be used to maintain the fixed length. *Contrast with*: **variable length.** *See also*: **fixed format.** [610.5]

fixed length field. A field whose length is constant. *Contrast with*: **variable length field.** *See also*: **fixed format.** [610.5]

fixed-point. Pertaining to a numeration system in which the position of the radix point is fixed with respect to one end of the numerals, according to some convention. *Contrast with:* **floating point; variable point.** [610.1, 610.5a]

fixed-point arithmetic. A method of arithmetic in which the numbers are expressed in the fixed-point representation system. *Contrast with:* **floating-point arithmetic.** [610.1]

fixed-point binary data. Fixed-point data used to represent signed binary numbers. *Syn*: **fixed binary; real fixed binary data.** [610.5]

decimal	75_{10}
fixed-point binary	$0100\ 1011_2$
decimal	-91_{10}
fixed point binary	$1010\ 0101_2$

fixed-point data. Integer data that can be expressed in a specific number of digits, with a radix point implicitly located at a predetermined position. *See also*: **fixed-point binary data; fixed-point real data.** *Contrast with*: **floating-point data.** *Syn*: **computational data.** [610.5]

fixed-point number. A number expressed in fixed-point representation. [610.1]

fixed-point part. *See:* **significand.** [610.1]

fixed-point real data. Fixed-point data used to represent signed decimal numbers. For example, 75.6, 0, and -253. *Syn*: **fixed decimal data; fixed real data; real fixed decimal data.** [610.5]

fixed-point representation system. A numeration system in which the position of the radix point is fixed with respect to one end of the numerals, according to some convention. [610.1]

fixed-radix notation. A radix notation system in which all digit positions have the same radix. The weights of successive digit places are successive integral powers of a single radix. *Syn:* **fixed-radix; numeration system; fixed-radix scale.** [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

fixed-radix numeration system. *See:* **fixed-radix notation.** [610.1]

fixed-radix scale. *See:* **fixed-radix notation.** [610.1]

fixed real data. *See:* **fixed-point real data.** [610.5]

flag. A variable that is set to a prescribed state, often "true" or "false," based on the results of a process or the occurrence of a specified condition. *See also:* **indicator; semaphore.** [610.12]

flash card. In micrographics, a target printed with distinctive markings to be photographed to facilitate the indexing of microfilm. *See also:* **flash indexing.** *Syn:* **flash target.** [610.2]

flash indexing. In micrographics, the process of dividing a roll of microfilm into batches of information using flash cards to identify each of the sections, thus providing a method of retrieval. *Syn:* **flash target coding.** [610.2]

flash target. *See:* **flash card.** [610.2]

flash target coding. *See:* **flash indexing.** [610.2]

flat file. (1) A set of records that are identically formatted to contain no more than one occurrence of each data item. *Note:* records in such a file do not contain data aggregates or repeating groups. *See also:* **relational file.** [610.5]
(2) A two-dimensional array of data items that is stored as in (1). [610.5]

flex model. *See:* **CODASYL model.** [610.5]

flexibility. The ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed. *Syn:* **adaptability.** *See also:* **extendability; maintainability.** [610.12]

flexible manufacturing system (FMS). A computer-integrated manufacturing system that can be reprogrammed to make a variety of parts or products. [610.2]

float. To convert a number from fixed-point representation to floating-point representation. *Contrast with:* **fix.** [610.1]

floating character. A character placed in the position that is one place more significant than the otherwise most significant position. [610.1]

floating decimal.* *See:* **floating point.** [610.1]
*Deprecated.

floating point. Pertaining to a numeration system in which each number is represented as a fractional quantity multiplied by an integral power of the radix. *Contrast with:* **fixed point; variable point.** [610.1, 610.5]

floating-point arithmetic. A method of arithmetic in which the numbers are expressed in the floating-point representation system. *Contrast with:* **fixed-point arithmetic.** [610.1]

floating-point coding compaction. A method of numerical data compaction that uses the floating-point representation system. [610.1]

floating-point coefficient. *See:* **exponent (2).** [610.1]

floating-point data. Real data in which numbers are represented using only an exponent, y, and a mantissa, x, where x and y are integers. *Note:* The number is expressed in the form $x \cdot 10^y$, and only x and y are stored in fixed-point binary format. *Syn:* **floating-point real data.** *Contrast with:* **fixed-point data.** [610.5]

decimal $12.3 = .123 \cdot 10^2$
floating-point $0111\ 1011\ 0000\ 0010_2 = 7B02_{16}$
----x---- ----y----

floating-point number. A digit string characterized by three components: a sign, a signed exponent, and a significand. Its numerical value, if any, is the signed product of its significand and the radix raised to the power of its exponent. [610.1]

floating-point real data. *See:* **floating-point data.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052832 8

floating-point representation system. A numeration system in which each number is represented as a sign, a signed exponent, and a significand, where the numerical value, if any, is the signed product of its significand and the radix raised to the power of the exponent. [610.1]

floor. The result obtained by rounding a number down to the nearest integer. For example, the floor of 5.3 is 5. *Contrast with:* **ceiling.** [610.1]

flowchart (flow chart). A control flow diagram in which suitably annotated geometrical figures are used to represent operations, data, or equipment, and arrows are used to indicate the sequential flow from one to another. *Syn:* **flow diagram.** *See also:* **block diagram; box diagram; bubble chart; graph; input-process-output chart; structure chart.** [610.12]

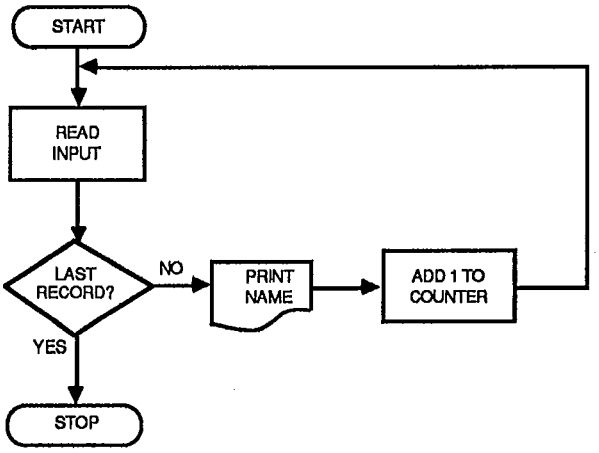


Fig 37
Flowchart

flowcharter. A software tool that accepts as input a design or code representation of a program and produces as output a flowchart of the program. [610.12]

flow diagram. *See:* **flowchart.** [610.12]

flow of control. *See:* **control flow.** [610.12]

flush left. In text formatting, justification of text such that it is aligned on the left and has a ragged right margin. *See also:* **left justification.** *Contrast with:* **flush right.** [610.2]

flush right. In text formatting, justification of text such that it is aligned on the right and has a ragged left margin. *See also:* **right justification.** *Contrast with:* **flush left.** [610.2]

FMS. Acronym for **flexible manufacturing system.** [610.2]

font disk. In phototypesetting, a glass disk, imprinted with a specific character font, used by a phototypesetter to generate characters in that character font. [610.2]

footer. *See:* **running footer.** [610.2]

forbidden character. *See*: **illegal character.** [610.5]

forbidden combination check. A check in which a combination of bits or other representations is not valid according to some criteria. *Contrast with*: **illegal character.** [610.5]

foreground. In job scheduling, the computing environment in which highpriority processes or those requiring user interaction are executed. *Contrast with:* **background.** *See also:* **foreground processing.** [610.12]

foreground processing. The execution of a high-priority process while lower-priority processes await the availability of computer resources, or the execution of processes that require user interaction. *Contrast with:* **background processing.** [610.12]

foreign key. (1) An attribute that is a primary key, not to the record it is in, but to some related record. [610.5]
(2) In a relational data model, nonprime attributes of some relation that is defined on the same domain as a prime attribute of another relation. [610.5]

forest. A set of disjoint trees. [610.5]

formal language. A language whose rules are explicitly established prior to its use. Examples include programming languages and mathematical languages. *Contrast with:* **natural language.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

formal parameter. A variable used in a software module to represent data or program elements that are to be passed to the module by a calling module. *Contrast with:* **argument (3).** [610.12]

formal qualification review (FQR). The test, inspection, or analytical process by which a group of configuration items comprising a system are verified to have met specific contractual performance requirements. *Contrast with:* **code review; design review; requirements review; test readiness review.** [610.12]

formal specification. (1) A specification written and approved in accordance with established standards. [610.12]
(2) A specification written in a formal notation, often for use in proof of correctness. [610.12]

formal testing. Testing conducted in accordance with test plans and procedures that have been reviewed and approved by a customer, user, or designated level of management. *Contrast with:* **informal testing.** [610.12]

format character. A control character used to control a printer. [610.5]

format effector character (FE). Any control character used to control the positioning of printed, displayed, or recorded data. *Syn:* **layout character.** *See also:* **back space character.** [610.5]

format status line. A line displayed by many word processing systems that shows the current setting of text formatting parameters such as tabulation stops and margin positions. [610.2]

format. The arrangement, order, or layout of data in or on a data medium. *See also:* **fixed format; variable format.** [610.5]

formatted information. Information that has been arranged into discrete units and structures in a manner that facilitates its access and processing. *Contrast with:* **narrative information.** [610.5]

formatted. (1) Pertaining to magnetic media, such as tapes or diskettes, that have been initialized and prepared to accept and store data. [610.2]
(2) Pertaining to text that has been organized into a particular arrangement for output or display. [610.2]

form feed character (FF). A format effector character that causes the print or display position to move to the next predetermined first line on the next form, the next page, or the equivalent. *Syn:* **page eject character; paper throw character.** [610.5]

form, fit, and function. In configuration management, that configuration comprising the physical and functional characteristics of an item as an entity, but not including any characteristics of the elements making up the item. *See also:* **configuration identification.** [610.12]

form letter. *See:* **iterative document.** [610.2]

forward recovery. (1) The reconstruction of a file to a given state by updating an earlier version, using data recorded in a chronological record of changes made to the file. *Contrast with:* **backward recovery; inline recovery.** [610.5, 610.12]
(2) A type of recovery in which a system, program, database, or other system resource is restored to a new, not previously occupied state in which it can perform required functions. [610.12]

four-address instruction. A computer instruction that contains four address fields. For example, an instruction to add the contents of locations A, B, and C, and place the result in location D. *Contrast with:* **one-address instruction; two-address instruction; three-address instruction; zero-address instruction.** [610.12]

four-bit byte. *See:* **quartet.** [610.5]

four-plus-one address instruction. A computer instruction that contains five address fields, the fifth containing the address of the instruction to be executed next. For example, an instruction to add the contents of locations A, B, and C, place the results in loca-

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

tion D, then execute the instruction at location E. *Contrast with:* **one-plus-one address instruction; two-plus-one address instruction; three-plus-one address instruction.** [610.12]

fourth generation language (4GL). A computer language designed to improve the productivity achieved by high-order (third generation) languages and, often, to make computing power available to non-programmers. Features typically include an integrated database management system, query language, report generator, and screen definition facility. Additional features may include a graphics generator, decision support function, financial modeling, spreadsheet capability, and statistical analysis functions. *Contrast with:* **machine language; assembly language; high-order language; fifth generation language.** *Note:* Specific languages will be defined in P610.13. [610.12]

fourth normal form (4NF). One of the forms used to characterize relations; a relation R is said to be in fourth normal form if it is in Boyce/Codd normal form and if, when there exists a non-trivial multivalued dependency A -> -> B, then all attributes in R are also functionally dependent on A. [610.5]

FQR. Acronym for **formal qualification review.** [610.12]

fraction. In floating point arithmetic, the component of the significand that lies to the right of its implied radix point. [610.1]

fractional binary. Pertaining to a binary numeral with the binary point (expressed or implied) at the left end, representing a fraction. [610.1]

fractional fixed point. Pertaining to fixed-point numeration system in which each number is represented by a numeral with the radix point (expressed or implied) at the left end. All numbers greater than or equal to one must be scaled accordingly. [610.1]

free-form typing. In word processing, the process of entering text that does not include text formatting commands. [610.2]

full backup. To perform a backup in which all data within a system is stored on the backup copy. *Contrast with:* **incremental backup.** [610.5]

full binary tree. *See:* **complete binary tree.** [610.5]

full functional dependency. A functional dependency in which no attribute of the determinant can be omitted without voiding the dependent condition. [610.5]

full justification. In text formatting, justification resulting in even margins on both the left and right margins. [610.2]

full-screen editing. A method of text editing that allows the user to view a full display screen of text at one time and to enter or alter text by using either commands or cursor control. Scrolling functions allow the user to move up and down within the document. *Contrast with:* **line editing.** [610.2]

full-screen editor. A text editor that allows the user to view a full display screen of data at one time and to enter or alter text by using either commands or cursor control. Scrolling functions allow the user to move up and down within the document. *Contrast with:* **line editor.** *Syn:* **screen editor.** [610.2]

full tree. *See:* **complete tree.** [610.5]

fullword. *See:* **word.** [610.5]

fully concatenated key. *See:* **concatenated key.** [610.5]

fully inverted file. An file that has been inverted on all secondary keys in the file. *Contrast with:* **partially inverted file.** [610.5]

fully relational. Pertaining to a database management system that supports a relational database and a language that provides the functionality of the relational algebra. [610.5]

function. (1) A defined objective or characteristic action of a system or component. For example, a system may have inventory control as its primary function. *See also:*

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

functional requirement; functional specification; functional testing. [610.12]
(2) A software module that performs a specific action, is invoked by the appearance of its name in an expression, may receive input values, and returns a single value. *See also:* **subroutine.** [610.12]
(3) A mathematical entity whose value is uniquely determined by the value of one or more independent variables. [610.2]

functional baseline. In configuration management, the initial approved technical documentation for a configuration item. *Contrast with:* **allocated baseline; developmental configuration; product baseline.** *See also:* **functional configuration item.** [610.12]

functional character. *See:* **control character.** [610.5]

functional cohesion. A type of cohesion in which the tasks performed by a software module all contribute to the performance of a single function. *Contrast with:* **coincidental cohesion; communicational cohesion; logical cohesion; procedural cohesion; sequential cohesion; temporal cohesion.** [610.12]

functional configuration audit (FCA). An audit conducted to verify that the development of a configuration item has been completed satisfactorily, that the item has achieved the performance and functional characteristics specified in the functional or allocated configuration identification, and that its operational and support documents are complete and satisfactory. *See also:* **configuration management; physical configuration audit.** [610.12]

functional configuration identification. In configuration management, the current approved technical documentation for a configuration item. It prescribes all necessary functional characteristics, the tests required to demonstrate achievement of specified functional characteristics, the necessary interface characteristics with associated configuration items, the configuration item's key functional characteristics and its key lower level configuration items, if any, and design constraints. *Contrast with:* **allocated configuration identification; product configuration identification.** *See also:* **functional baseline.** [610.12]

functional decomposition. A type of modular decomposition in which a system is broken down into components that correspond to system functions and subfunctions. *See also:* **hierarchical decomposition; stepwise refinement.** [610.12]

functional dependency. A type of dependency between two attributes A and B in a relation, in which B is functionally dependent on A if, and only if, at every instant in time, each value of A is associated with no more than one value of B. *Note*: A is said to "identify" or "functionally determine" B. Written A -> B. *See also*: **full functional dependency; join dependency.** [610.5]

functional design. (1) The process of defining the working relationships among the components of a system. *See also:* **architectural design.** [610.12]
(2) The result of the process in (1). [610.12]

functional language. A programming language used to express programs as a sequence of functions and function calls. Examples include LISP. [610.12]

functional requirement. A requirement that specifies a function that a system or system component must be able to perform. *Contrast with:* **design requirement; implementation requirement; interface requirement; performance requirement; physical requirement.** [610.12]

functional specification. A document that specifies the functions that a system or component must perform. Often part of a requirements specification. [610.12]

functional testing. (1) Testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions. *Syn:* **blackbox testing.** *Contrast with:* **structural testing.** [610.12]
(2) Testing conducted to evaluate the compliance of a system or component with specified functional requirements. *See also:* **performance testing.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

functionally determined. *See*: **functional dependency.** [610.5]

function field. *See:* **operation field.** [610.12]

G. Abbreviation for **giga.** [610.1]

game. A physical or mental competition in which the participants, called players, seek to achieve some objective within a given set of rules. *See also:* **game theory.** [610.3]

game theory. (1) The study of situations involving competing interests, modeled in terms of the strategies, probabilities, actions, gains, and losses of opposing players in a game. *See also:* **management game; war game.** [610.3]
(2) The study of games to determine the probability of winning given various strategies. [610.3]

gaming simulation. *See:* **simulation game.** [610.3]

gap character. A character that is included in a computer word for technical reasons but that does not represent data. [610.5]

garbage. Unwanted or meaningless data. [610.5]

garbage collection. (1) In data management, a space optimization technique in which superfluous data are eliminated. [610.5]
(2) In data management, a database reorganization technique in which the contents of a database are made more compact by physically deleting garbage such as records that have been deleted logically but remain physically in the database. [610.5]
(3) In computer resource management, a synonym for **memory compaction (1).** [610.12]

GDG. Acronym for **generation data group.** [610.5]

generality. The degree to which a system or component performs a broad range of functions. *See also:* **reusability.** [610.12]

generated address. An address that has been calculated during the execution of a computer program. *Syn:* **synthetic address.** *See also:* **absolute address; effective address; relative address; indirect address.** [610.12]

generated error. The total error resulting from the combined effects of using imprecise arguments in an inexact formula. For example, using a rounded number in a truncated series. [610.1]

generation data group (GDG). A collection of data files that are kept in chronological order and referenced by its generation number. *Note:* Each file is called a generation data set. [610.5]

generation data set. One data file within a generation data group. [610.5]

generic data element. A data element related to or drawn from a large class of like data elements. [610.5]

generic program unit. A software module that is defined in a general manner and that requires substitution of specific data, instructions, or both in order to be used in a computer program. *See also*: **instantiation.** [610.12]

geometric correction. An image restoration technique in which a geometrical transformation is performed on an image to compensate for geometrical distortions. [610.4]

get. (1) To retrieve an item from a set of items as in retrieving a record from a file, or in obtaining a numerical value from a series of decimal digits. *Contrast with*: **put.** [610.5]
(2) To select and retrieve a group of specified records from a database. [610.5]

get next. To select and retrieve the next record from a database that meets some specified criteria. *Note*: Used in conjunction with a placeholder point. *Contrast with*: **get unique.** [610.5]

get unique. To select and retrieve the first record from a database that meets some selection criteria. *Contrast with*: **get next.** [610.5]

ghost hyphen. *See:* **discretionary hyphen.** [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

giga (G). A prefix indicating 10^9. [610.1a]

glass box. (1) A system or component whose internal contents or implementation are known. *Syn:* **white box.** *Contrast with:* **black box.** [610.12]
(2) Pertaining to an approach that treats a system or component as in (1). [610.12]

glass box model. A model whose internal implementation is known and fully visible; for example, a model of a computerized change-return mechanism in a vending machine, in the form of a diagram of the circuits and gears that make the change. *Contrast with:* **black box model.** *Syn:* **white box model.** [610.3]

glass-box testing. *See:* **structural testing.** [610.12]

global compaction. In microprogramming, compaction in which microoperations may be moved beyond the boundaries of the single entry, single exit sequential blocks in which they occur. *Contrast with:* **local compaction.** [610.12]

global data. Data that can be accessed by two or more non-nested modules of computer program without being explicitly passed as parameters between the modules. *Syn:* **common data.** *Contrast with:* **local data.** [610.12]

global replace. In text editing, an operation that substitutes a given textual pattern for all, or a given number of, occurrences of some other textual pattern found in the text. *See also:* **global search.** [610.2]

global search. In text editing, an operation that identifies all, or a given number of, appearances of a given textual pattern in the text. *See also:* **global replace.** [610.2]

global variable. A variable that can be accessed by two or more non-nested modules of a computer program without being explicitly passed as a parameter between the modules. *Contrast with:* **local variable.** [610.12]

glossary. *See:* **data glossary.** [610.5]

go list. In automatic indexing, a list of terms, words, or roots of words that are considered significant for purposes of information retrieval, and are to be used as keywords in an index. *Contrast with:* **stop list.** *Syn:* **inclusion list.** [610.2]

go to. A computer program statement that causes a jump. *Contrast with:* **call; case; if-then-else.** *See also:* **branch.** [610.12]

grandfather file. A file that contains data that have since been updated in another file, called the father file, and further updated in a third file, called the son file. [610.5]

graph. (1) A diagram that represents the variation of a variable in comparison with that of one or more other variables; for example, a graph showing a bathtub curve. [610.12]

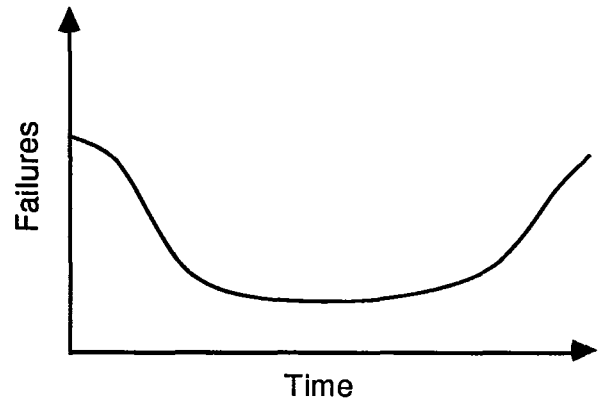


Fig 38
Graph (1)

(2) A diagram or other representation consisting of a finite set of nodes and internode connections called edges or arcs. *See also:* **block diagram; box diagram; bubble chart; directed graph; flowchart; input-process-output chart; structure chart; undirected graph.** [610.5, 610.12]

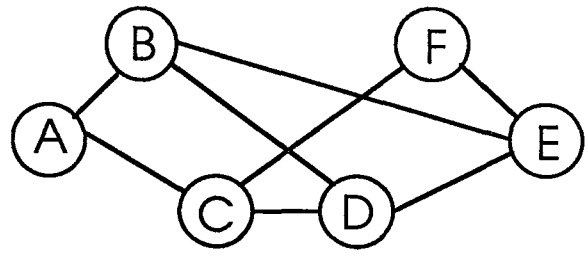


Fig 39
Graph (2)

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

graphic. A symbol produced by a process such as handwriting, drawing, or printing. *Syn:* **graphic symbol.** [610.2]

graphical model. A symbolic model whose properties are expressed in diagrams; for example, a decision tree used to express a complex procedure. *Contrast with:* **mathematical model; narrative model; software model; tabular model.** [610.3]

graphic character. A character, other than a control character, that is normally represented by a graphic. *Syn:* **optical character.** [610.2]

graphic symbol. *See:* **graphic.** [610.2]

Gray code. A binary code in which sequential numbers are represented by binary expressions, each of which differs from the preceding expression in one place only. *Syn:* **cyclic binary code; cyclic code; reflected code; reflected-binary code; reflected-binary unit-distance code.** [610.1]

DECIMAL DIGIT:	0	1	2	3	4	5
GRAY CODE:	000	001	011	010	110	111

Fig 40
Example of a Gray Code

gray level. A value associated with a pixel in a digital image, representing the brightness of the original scene in the vicinity of the point represented by the pixel. *Syn:* **gray shade; gray tone.** [610.4]

gray scale. The range of gray levels that occur in an image. [610.4]

gray scale manipulation. An image enhancement technique in which the appearance of a digital image is improved by applying a point operator to each pixel in the image, adjusting its gray level. [610.4]

gray shade. *See:* **gray level.** [610.4]

gray tone. *See:* **gray level.** [610.4]

grid. In optical character recognition, two perpendicular sets of parallel lines used for specifying or measuring character images. [610.2]

Grosch's law. A guideline formulated by H. R. J. Grosch, stating that the computing power of a computer increases proportionally to the square of the cost of the computer. *See also:* **computer performance evaluation.** [610.12]

group. A set of items that are related to each other in some way; for example, a set of records that have the same value for a particular field, or a set of files in a generation data group. [610.5]

group item. *See:* **data aggregate.** [610.5]

guide edge. *See:* **document edge.** [610.2]

half-adjust. To round a number by changing the least significant digit to zero and adding one to the next digit if the value of the least significant digit was half the radix or greater. [610.1]

half carry. A carry process in which a carry digit generated in the most significant digit place of the less-significant half of a sum is transferred to the least significant digit place of the more significant half. [610.1]

halt. (1) Most commonly, a synonym for **stop.** [610.12]
(2) Less commonly, a synonym for **pause.** [610.12]

Hamming code. Any of several error-correcting codes invented by the mathematician Richard Hamming, which use redundant information bits to detect and correct any single error in a transmitted character. *See also:* **error-correcting code.** [610.1]

Hamming distance. The number of digit positions in which two binary numerals, characters, or words of the same length are different. For example, the Hamming distance between 100101 and 101001 is two. *Syn:* **code distance; signal distance.** [610.1]

hand-printed character font. An international standard optical font for use on hand-generated documents. *See also:* **OCR-A; OCR-B.** [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

hand-print recognition. Optical character recognition of hand-printed characters. [610.2]

hard copy. A printed copy of computer output in a visually readable form; for example, a printed report, a listing. *Contrast with:* **soft copy.** [610.2]

hard failure. A failure that results in complete shutdown of a system. *Contrast with:* **soft failure.** [610.12]

hardware. Physical equipment used to process, store, or transmit computer programs or data. *Contrast with:* **software.** [610.12]

hardware check. *See*: **automatic check.** [610.5]

hardware configuration item (HWCI). An aggregation of hardware that is designated for configuration management and treated as a single entity in the configuration management process. *Contrast with:* **computer software configuration item.** *See also:* **configuration item.** [610.12]

hardware design language (HDL). A language with special constructs and, sometimes, verification protocols, used to develop, analyze, and document a hardware design. *See also:* **program design language.** [610.12]

hardware monitor. (1) A device that measures or records specified events or characteristics of a computer system; for example, a device that counts the occurrences of various electrical events or measures the time between such events. [610.12]
(2) A software tool that records or analyzes hardware events during the execution of a computer program. [610.12]
See also: **monitor; software monitor.** [610.12]

hash. To calculate the hash value for a given item. *See also*: **hashing.** [610.5]

hash address. *See*: **hash value.** [610.5]

hash addressing. *See*: **hashing.** [610.5]

hash clash. *See*: **collision.** [610.5]

hash coding. *See*: **hashing.** [610.5]

hash function. In hashing, the function used to determine the position of a given item in a set of items. *Note*: The function operates on a selected field, called a key, in each item and the function is generally a many-to-one mapping. *Syn*: **calc algorithm; key transformation function.** *See also*: **algebraic coding function; digit transformation function; division transformation function; key folding function; key transformation; mid-square function; multiplication transformation function; radix transformation function.** [610.5]

hash index. *See*: **hash value.** [610.5]

hashing. A technique for arranging a set of items, in which a hash function is applied to the key of each item to determine its hash value. The hash value identifies each item's primary position in a hash table, and if this position is already occupied, the item is inserted either in an overflow table or in another available position in the table. *Syn*: **hash coding; hash-addressing; randomizing; scatter storage.** *See also*: **collision resolution; open-address hashing; separate chaining.** [610.5]

hash search. The use of a hash function and collision resolution to locate an item in a hash table. [610.5]

hash table. A two-dimensional table of items in which a hash function is applied to the key of each item to determine its hash value. The hash value identifies each item's primary position in the table, and if this position is already occupied, the item is inserted either in an overflow table or in another available position in the table. [610.5]

hash total. The result of summing two or more values of a set for purposes of validation or error detection. *Syn*: **control total.** [610.5]

hash value. The number generated by a hash function to indicate the position of a given item in a hash table. *Syn*: **hash address; hash index.** [610.5]

HDAM. Acronym for **hierarchical direct access method.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

HDL. Acronym for **hardware design language.** *See:* **design language.** [610.12]

head. The first data item in a list. *Syn*: **header (2).** [610.5]

header. (1) A block of comments placed at the beginning of a computer program or routine. [610.12]
(2) Pertaining to data that describes and pertains to other data. For example, the header record for a file might describe the format for the remaining records in the file. [610.5]
(3) Identification or control information placed at the beginning of a file or message. *Contrast with:* **trailer.** [610.12]
(4) *See*: **head.** [610.5]
(5) *See:* **running header.** [610.2]

header label. An internal label, immediately preceding the first record of a file, that identifies the file and contains data used in file control. [610.5]

health information system (HIS). *See:* **hospital information system.** [610.2]

heap. A complete binary tree in which the key for each child node contains the key from its parent plus some additional value. [610.5]

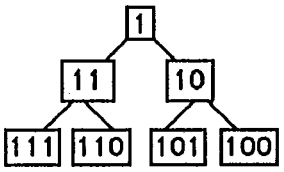


Fig 41
Heap

heapsort (heap sort). A tree selection sort in which the items to be sorted are used to build a heap, and the items are then selected from the heap in the sorted order. [610.5]

height. In a tree, the maximum number of levels between the root node and a terminal node. *See also*: **height-balanced tree.** *Syn*: **depth.** [610.5]

height balance. In a tree, the maximum difference in height of any two subtrees of any node. *Note:* A height balance of k is written HB-k. [610.5a]

height-balanced k-tree. A tree whose height balance is k. [610.5]

height-balanced tree. A tree whose height balance is 1. *See also*: **Adel'son-Velskii and Landis (AVL) tree; B-tree; n-m tree.** *Contrast with*: **weight-balanced tree.** *Syn*: **balanced tree.** [610.5]

help. *See:* **help information.** [610.2]

help file. A file containing help information. [610.5]

help information. Information available for display to the user of a computer system, describing system features and use. *Syn:* **help.** *See also:* **help menu.** [610.2]

help menu. A menu that gives the user a choice of topics for which help information is available on a given computer system. [610.2]

heuristic. Pertaining to experimental, especially trial-and-error, methods of problem-solving. *Note:* The resulting solution may not be the most desirable solution to the problem. [610.3]

hex. Abbreviation for **hexadecimal.** [610.1a]

hexadecimal. (1) Pertaining to a selection in which there are sixteen possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of 16. *Syn:* **sexadecimal.** [610.1]

hexadecimal digit. A numeral used to represent one of the 16 digits in the hexadecimal numeration system; 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, or F. [610.1]

hexadecimal notation. Any notation that uses the hexadecimal digits and the radix 16. [610.1]

hexadecimal number. (1) A quantity that is expressed using the hexadecimal numeration system. [610.1]
(2) Loosely, a hexadecimal numeral. [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

hexadecimal number system. *See:* **hexadecimal numeration system.** [610.1]
*Deprecated.

hexadecimal numeral. A numeral in the hexadecimal numeration system. For example, the hexadecimal numeral 17 is equivalent to the decimal numeral 23. [610.1]

hexadecimal numeration system. The numeration system that uses the hexadecimal digits and the radix 16. *Syn:* **hexadecimal system.** [610.1]

hexadecimal point. The radix point in the hexadecimal numeration system. [610.1]

hexadecimal system. *See:* **hexadecimal numeration system.** [610.1]

hexadecimal-to-decimal conversion. The process of converting a hexadecimal numeral to an equivalent decimal numeral. For example, hexadecimal 8B.4 is converted to decimal 139.25. [610.1]

HIDAM. Acronym for **hierarchical indexed direct access method.** [610.5]

hierarchical. Pertaining to a hierarchy, as in a hierarchical database or a hierarchical structure. [610.5]

hierarchical database. A database in which data are organized into records, known as segments, that represent nodes in a hierarchy or tree structure. *Note*: Within the hierarchy, a subordinate to a given segment is known as its child segment and a superordinate is known as its parent segment. *Contrast with*: **network database; relational database.** *Syn*: **sequential precedential database.** [610.5]

hierarchical decomposition. A type of modular decomposition in which a system is broken down into a hierarchy of components through a series of top-down refinements. *See also:* **functional decomposition; stepwise refinement.** [610.12]

hierarchical direct access method (HDAM). A database access method for hierarchical databases in which pointers maintain the structure itself as well as the control of the storage and retrieval functions of the database. All records are stored and retrieved using these pointers. *Contrast with*: **hierarchical sequential access method.** *See also*: **hierarchical indexed direct access method; hierarchical indexed sequential access method.** [610.5]

hierarchical indexed direct access method (HIDAM). A database access method for hierarchical databases in which indices access root segments and pointers access dependent segments. *Contrast with*: **hierarchical indexed sequential access method.** [610.5]

hierarchical indexed sequential access method (HISAM). A database access method for hierarchical databases in which indices control access to both root and dependent segments. *Contrast with*: **hierarchical indexed direct access method.** [610.5]

hierarchical input-process-output (HIPO). *See:* **input-process-output.** [610.12]

hierarchical model. (1) A data model whose pattern of organization is in the form of a tree structure. [610.5]
(2) A data model that provides a tree structure for relating data elements, where each node of the tree corresponds to a group of data elements or a record type, and has only one superior node or parent. [610.5]

hierarchical modeling. A technique used in computer performance evaluation, in which a computer system is represented as a hierarchy of subsystems, the subsystems are analyzed to determine their performance characteristics, and the results are used to evaluate the performance of the overall system. [610.12]

hierarchical sequence. In a hierarchical database, the sequence of root and dependent segments defined by traversing the database in some specified order. [610.5]

hierarchical sequential access method (HSAM). A database access method for hierarchical databases in which data items are stored and retrieved sequentially. *See*

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052842 0

also: **hierarchical indexed sequential access method.** *Contrast with*: **hierarchical direct access method.** [610.5]

hierarchical structure. A collection of entities that are organized in a hierarchical fashion. *Contrast with*: **network structure.** [610.5]

hierarchy. A structure in which components are ranked into levels of subordination; each component has zero, one, or more subordinates; and no component has more than one superordinate component. *See also*: **data hierarchy; hierarchical database; hierarchical decomposition; hierarchical modeling; link; network; tree.** [610.5, 610.12]

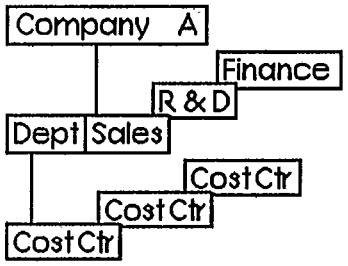


Fig 42
Hierarchy

hierarchy chart. *See:* **structure chart.** [610.12]

high-emphasis filtering. In image processing, a sharpening technique in which rapid fluctuations in gray levels are emphasized. [610.4]

higher-order language. *See:* **high-order language.** [610.12a]

high level language. *See:* **high-order language.** [610.12]

high-order. Pertaining to the left-most digit or digits of a numeral. [610.1]

high-order language (HOL). A programming language that requires little knowledge of the computer on which a program will run, can be translated into several different machine languages, allows symbolic naming of operations and addresses, provides features designed to facilitate expression of data structures and program logic, and usually results in several machine instructions for each program statement. Examples include Ada, COBOL, FORTRAN, ALGOL, PASCAL. *Syn:* **high level language; higher order language; third generation language.** *Contrast with:* **assembly language; fifth generation language; fourth generation language; machine language.** *Note:* Specific languages will be defined in P610.13. [610.12a]

high-order position. The leftmost position in a string; for example, the letter 'A' in 'APPLE' or the digit 9 in 965. *Contrast with*: **low-order position.** *See also*: **most significant character; most significant digit.** [610.5a]

high-speed carry. A carry process in which, if the current sum in a given digit place is one less than the base, the sum is set to zero and the carry input is passed to the next place. *Contrast with:* **cascaded carry.** *See also:* **standing-on-nines carry.** [610.1]

HIS. (1) Acronym for **hospital information system.** [610.2]
(2) Acronym for health information system. *See:* **hospital information system.** [610.2]

HISAM. Acronym for **hierarchical indexed sequential access method.** [610.5]

hit. (1) In a search, the condition that occurs when the key value of an item is equal to the search argument; that is, a successful search results in a hit. [610.5]
(2) A record that produces the condition in (1); for example, a student record in which the student's home state matches the home state being searched for. *Contrast with*: **match.** [610.5]

hit file. A file containing all records that resulted from a successful search. [610.5]

hit ratio. In a search, the number of hits divided by the total number of items searched. [610.5]

HLL. Acronym for **high level language.** *See:* **high-order language.** [610.12]

HMI. Acronym for **human-machine interface.** *See:* **user interface.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

HOL. Acronym for **high-order language.** [610.12]

hold. When performing a get operation on a record, to lock the record for update by the requesting process at the same time the get operation is performed. [610.5]

hole. In image processing, a connected component of the complement of a region, that is surrounded by the region. [610.4]

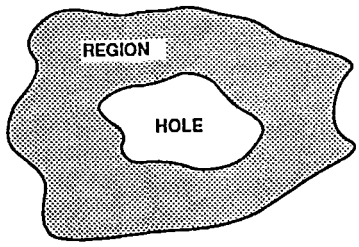


Fig 43
Illustration of Hole

home computer. A personal computer designed to be used in the home. [610.2]

homogeneous redundancy. In fault tolerance, realization of the same function with identical means, for example, use of two identical processors. *Contrast with:* **diversity.** [610.12]

horizontal microinstruction. A microinstruction that specifies a set of simultaneous operations needed to carry out a given machine language instruction. *Note:* Horizontal microinstructions are relatively long, often 64 bits or more, and are called "horizontal" because the set of simultaneous operations that they specify are written on a single line, rather than being listed sequentially down the page. *Contrast with:* **diagonal microinstruction; vertical microinstruction.** [610.12]

horizontal tabulation character (HT). A format effector character that causes the print or display position to move forward to the next of a series of predetermined positions along the same horizontal line. [610.5]

hospital information system (HIS). An automated system used in hospitals and other health care facilities to perform such tasks as communication between staff members, statistical analysis, inventory planning, and scheduling of medication, blood analysis, and patient testing. *Note:* Hospital information systems typically use interactive operations on a hierarchical file structure based on a patient-oriented record. *Syn:* **health information system; medical information system.** [610.2]

host language. A programming language such as COBOL or PL/I into which data manipulation language statements are embedded. *See also:* **data sublanguage.** [610.5]

host machine. (1) A computer used to develop software intended for another computer. *Contrast with:* **target machine (1).** [610.12]
(2) A computer used to emulate another computer. *Contrast with:* **target machine (2).** [610.12]
(3) The computer on which a program or file is installed. [610.12]
(4) In a computer network, a computer that provides processing capabilities to users of the network. [610.12]

hot zone. In text formatting, a predefined region at the right end of each line of text, having the characteristic that any word that begins in the region and extends beyond it is automatically moved to the next line, and any word that begins before the region and extends beyond it must be hyphenated. *Syn:* **line-end zone; line-ending zone; margin-adjust zone.** [610.2]

hot zone hyphenation. In text formatting, semi-manual hyphenation in which any word that extends beyond the hot zone must be either hyphenated or moved to the next line. [610.2]

housekeeping operation. A computer operation that establishes or reestablishes a set of initial conditions to facilitate the execution of a computer program; for example, initializing storage areas, clearing flags, rewinding tapes, opening and closing files. *Syn:* **overhead operation.** [610.12]

HSAM. Acronym for **hierarchical sequential access method.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

HT. Acronym for the **horizontal tabulation character.** [610.5]

human-centered simulation (human-centred simulation). A simulation carried out by people; for example, a simulation in which a human participant operates a mock-up of an instrument to establish a good ergonomic design of the instrument console. *Contrast with:* **computer-based simulation.** *See also:* **human-machine simulation.** [610.3]

human-machine interface (HMI). *See:* **user interface.** [610.12]

human-machine simulation. A simulation carried out by both human participants and computers, typically with the human participants asked to make decisions and a computer performing processing based on those decisions; for example, a simulation in which humans make automotive design decisions and a computer determines and displays the results of those decisions. [610.3]

HWCI. Acronym for **hardware configuration item.** [610.12]

hybrid coupling. A type of coupling in which different subsets of the range of values that a data item can assume are used for different and unrelated purposes in different software module. *Contrast with:* **common-environment coupling; content coupling; control coupling; data coupling; pathological coupling.** [610.12]

hybrid simulation. A simulation, portions of which are designed to be executed on an analog system and portions on a digital system. Interaction between the two portions may take place during execution. *See also:* **analog simulation; digital simulation.** [610.3]

hyphenation. In text formatting, a manual, semi-manual, or fully automatic process of selecting appropriate word breaks at the end of a line of text and inserting a hyphen at one of those breaks. *See also:* **automatic hyphenation; manual hyphenation; semi-manual hyphenation.** *Contrast with:* **hyphenless justification.** [610.2]

hyphen drop. In word processing, the automatic omission of a discretionary hyphen from formatted text when the hyphen is not needed to achieve justification. [610.2]

hyphenless justification. In text formatting, justification in which any word that will not fit entirely on one line is moved to the next line, and intercharacter or interword spacing is used to justify the text. *Contrast with:* **hyphenation.** [610.2]

IC. Acronym for **information center.** [610.2]

iconic model. A physical model that looks like the system being modeled; for example, a non-functional replica of a computer tape drive used for display purposes. *See also:* **scale model.** [610.3]

identified. *See:* **functional dependency.** [610.5]

identifier. The name, address, label, or distinguishing index of an object in a computer program. [610.12]

identity operation. A Boolean operation whose result is true if and only if the operands are all true or all false. *Note:* An identity operation on two operands is the same as an equivalence operation. [610.1]

identity simulation. A simulation in which the roles of the participants are investigated or defined; for example, a simulation that identifies aircraft based on their physical profiles, speed, altitude, and acoustic characteristics. [610.3]

idle. Pertaining to a system or component that is operational and in service, but not in use. *See also:* **busy; down; up.** [610.12]

idle character. A control character that is sent when there is no information to be sent. [610.5]

idle time. The period of time during which a system or component is operational and in service, but not in use. *Syn:* **standby time.** *See also:* **busy time; down time; set-up time; up time.** [610.12]

IDP. Acronym for **integrated data processing.** [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IF-AND-ONLY-IF. *See:* **equivalence.** [610.1]

IF-THEN. *See:* **implication.** [610.1]

if-then-else. A single-entry, single-exit two-way branch that defines a condition, specifies the processing to be performed if the condition is met and, optionally, if it is not, and returns control in both instances to the statement immediately following the overall construct. *Contrast with:* **case; jump; go to.** *See also:* **dyadic selective construct; monadic selective construct.** [610.12]

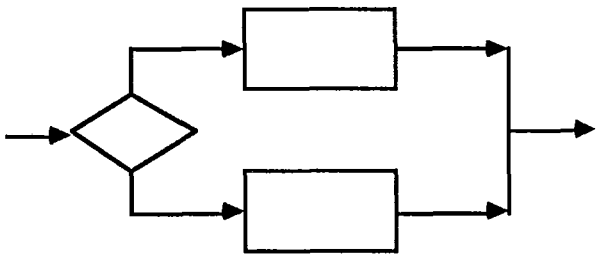


Fig 44
If-Then-Else Construct

illegal character. A character or combination of bits that is not valid according to some criteria; for example, a character that is not a member of some specified alphabet. *Contrast with:* **forbidden combination.** *Syn:* **forbidden character; improper character.** [610.5]

image. In image processing, a two-dimensional representation of a scene. *Syn:* **picture.** *See also:* **digital image.** [610.4]

image analysis. The process of describing or evaluating an image in terms of its parts, properties, and relationships. [610.4]

image area. In micrographics, that part of the film frame reserved for an image. [610.2]

image compression. The process of eliminating redundancy or approximating an image in order to represent the image in a more compact manner. *See also:* **adaptive coding; contour encoding; interframe coding; lossless encoding; predictive coding; run length encoding.** [610.4]

image dissector. In optical character recognition, a mechanical or electronic device that sequentially detects the level of light intensity in different areas of a completely illuminated sample space. [610.2]

image enhancement. The process of improving the appearance of an image by using techniques such as contrast stretching, edge enhancement, gray scale manipulation, smoothing, and sharpening. [610.4]

image matching. An image processing technique in which similar patterns are detected by comparing corresponding points of two images. *See also:* **template matching.** [610.4]

image operator. A function that transforms an input image into an output image. *Syn:* **image transform; image transform operator.** *See also:* **neighborhood operator; point operator.** [610.4]

image processing. The manipulation of images by computer. *Syn:* **picture processing.** *See also:* **change detection; congruencing; edge linking; sampling; thinning.** [610.4]

image reconstruction. The process of recovering an image from integrals of its gray levels taken along thin strips or slices of the image. [610.4]

image registration. The process of positioning two images of the same scene with respect to one another so that corresponding points in the images represent the same point in the scene. *See also:* **registered images.** [610.4]

image restoration. The process of returning an image to its original condition by reversing the effects of known or estimated degradations. *See also:* **geometric correction.** [610.4]

image segmentation. The process of dividing an image into regions. *Syn:* **object extraction.** *See also:* **border detection; edge detection; line detection; region growing; tracking.** [610.4]

image transform operator. *See:* **image operator.** [610.4]

image transform. *See:* **image operator.** [610.4]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052846 8

immediate access. Access to a storage device or register in which access time is virtually equal to zero. *Note*: Access time measured in nanoseconds is considered to be virtually equal to zero. *Syn*: **instantaneous access; simultaneous access.** [610.5]

immediate address.* *See:* **immediate data.**
* Deprecated. [610.12]

immediate control. *See:* **bit steering.** [610.12]

immediate data. Data contained in the address field of a computer instruction. *Contrast with:* **direct address; indirect address; *n*-level address.** *See also:* **immediate instruction.** [610.12]

immediate instruction. A computer instruction whose address fields contain the values of the operands rather than the operands' addresses. *Contrast with:* **direct instruction; indirect instruction.** *See also:* **absolute instruction; effective instruction; immediate data.** [610.12]

imperative construct. A sequence of one or more steps not involving branching or iteration. [610.12]

imperative statement. *See:* **instruction.** [610.12]

implementation. (1) The process of translating a design into hardware components, software components, or both. *See also:* **coding.** [610.12]
(2) The result of the process in (1). [610.12]

implementation design. *See*: **database design.** [610.5]

implementation phase. The period of time in the software life cycle during which a software product is created from design documentation and debugged. [610.12]

implementation requirement. A requirement that specifies or constrains the coding or construction of a system or system component. *Contrast with:* **design requirement; functional requirement; interface requirement; performance requirement; physical requirement.** [610.12]

implication. A dyadic Boolean operator having the property that if P is a statement and Q is a statement, then the expression "P implies Q" is true in all cases except when P is true and Q is false. *Note:* P implies Q is often represented as P→Q. *Syn:* **conditional implication; IF-THEN.** [610.1]

P	Q	P→Q
0	0	1
0	1	1
1	0	0
1	1	1

Fig 45
Implication Truth Table

implied addressing. A method of addressing in which the operation field of an computer instruction implies the address of the operands. For example, if a computer has only one accumulator, an instruction that refers to the accumulator needs no address information describing it. Types include one-ahead addressing, repetitive addressing. *See also:* **direct address; indirect address; relative address.** [610.12]

implied binary point. *See:* **assumed binary point.** [610.1]

implied decimal point. *See:* **assumed decimal point.** [610.1]

implied radix point. *See:* **assumed radix point.** [610.1]

improper character. *See*: **illegal character.** [610.5]

in-basket simulation. A simulation in which a set of issues is presented to a participant in the form of documents on which action must be taken; for example, a simulation of an unfolding international crisis as a sequence of memos describing relevant events and outcomes of the participant's actions on previous memos. [610.3]

inactive. (1) Pertaining to a record or file that has not been accessed by an update transaction during a given processing cycle. [610.2]
(2) Pertaining to a record that will not be processed by future transactions. *See also:* **active; purged; logically deleted.** [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

incident. *See:* **software test incident.** [610.12]

incipient failure. A failure that is about to occur. [610.12]

inclusion list. *See:* **go list.** [610.2]

inclusion. *See:* **implication.** [610.1]
*Deprecated.

inclusive OR. *See:* **OR.** [610.1]

in-core sort. *See:* **internal sort.** [610.5]

increment. (1) The quantity by which a variable is increased. [610.1]
(2) To increase the value of a variable. *Contrast with:* **decrement (2).** [610.1]
(3) To increase the value of a variable by one. *Contrast with:* **decrement (3).** [610.1]

incremental backup. To perform a backup of a system in which the only data that is stored on the backup is data that has been modified since the last full backup was performed. *Contrast with:* **full backup.** [610.5]

incremental binary representation. *See:* **binary incremental representation.** [610.1]

incremental compiler. A compiler that completes as much of the translation of each source statement as possible during the input or scanning of the source statement. Typically used for on-line computer program development and checkout. *Syn:* **conversational compiler; interactive compiler; on-line compiler.** [610.12]

incremental development. A software development technique in which requirements definition, design, implementation, and testing occur in an overlapping, iterative (rather than sequential) manner, resulting in incremental completion of the overall software product. *Contrast with:* **waterfall model.** *See also:* **data structure-centered design; input-process-output; modular decomposition; object-oriented design; rapid prototyping; spiral model; stepwise refinement; structured design; transaction analysis; transform analysis.** [610.12]

incremental justification. In text formatting, the use of extra intercharacter spacing to form even margins. *Contrast with:* **line filling.** [610.2]

incremental representation. Any number representation system in which the numerals express changes in the variables rather than the variables themselves. *See also:* **binary incremental representation; ternary incremental representation.** [610.1]

incremental ternary representation. *See:* **ternary incremental representation.** [610.1]

independent variable. A variable whose value is not dependent on the values of other variables. *Contrast with:* **dependent variable.** [610.3]

independent verification and validation (IV&V). Verification and validation performed by an organization that is technically, managerially, and financially independent of the development organization. [610.12]

index. (1) A data item that identifies a particular element in a set of items such as an array. [610.5]
(2) A list or table used to locate records within an indexed file that contains the location and unique key value of each record. *Syn:* **directory.** *See also:* **alternate index; cross-index.** [610.5a]
(3) To prepare a table as in (2). [610.5]

indexed access. The process of accessing stored data in such a way that indices are used to locate records within data storage. *Syn:* **keyed access.** *See also:* **indexed sequential access.** [610.5]

indexed address. An address that must be added to the contents of an index register to obtain the address of the storage location to be accessed. *See also:* **offset (2); relative address; self-relative address.** [610.12]

indexed file. A file that may be accessed using an index. *Contrast with:* **partitioned data set; sequential file.** [610.5]

indexed segment. In a database, a segment that is located by an indexing segment. *Syn:* **index target segment.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

indexed sequential access method (ISAM). An access method by which data records may be stored and retrieved using either the sequential access method or the direct access method. *See also*: **basic sequential access method; queued sequential access method; virtual sequential access method.** [610.5]

indexed sequential access mode. An access mode in which data records may be stored and retrieved using either direct access mode or sequential access mode. *Note*: The records are actually stored in a sequential fashion, but an index is maintained to allow direct access. *Contrast with*: **direct access mode; sequential access mode.** [610.5]

indexed sequential access. The process of accessing stored data using the indexed sequential access mode. *Contrast with*: **direct access; sequential access.** *See also*: **indexed access.** [610.5]

indexing segment. In a database, a segment that contains a pointer to another segment, called the indexed segment, containing data. *Syn*: **index pointer segment.** [610.5]

index pointer segment. *See*: **indexing segment.** [610.5]

index target segment. *See*: **indexed segment.** [610.5]

indicator. A device or variable that can be set to a prescribed state based on the results of a process or the occurrence of a specified condition. For example, a flag or semaphore. [610.12]

indices. Plural form of index. [610.5]

indigenous error. A computer program error that has not been purposely inserted as part of an error-seeding process. [610.12]

indirect address. An address that identifies the storage location of another address. The designated storage location may contain the address of the desired operand or another indirect address; the chain of addresses eventually leads to the operand. *Syn:* **multilevel address.** *Contrast with:* **direct address; immediate data.** *See also:* **indirect instruction; *n*-level address.** [610.12]

indirect instruction. A computer instruction that contains indirect addresses for its operands. *Contrast with:* **direct instruction; immediate instruction.** *See also:* **absolute instruction; effective instruction.** [610.12]

inductive assertion method. A proof of correctness technique in which assertions are written describing program inputs, outputs, and intermediate conditions, a set of theorems is developed relating satisfaction of the input assertions to satisfaction of the output assertions, and the theorems are proved or disproved using proof by induction. [610.12]

inequivalence. *See:* **exclusive OR.** [610.1]

infant mortality. The set of failures that occur during the early-failure period of a system or component. [610.12]

infix notation. A method of forming mathematical expressions in which each operator is written between its operands and the expression is interpreted subject to rules of operator precedence and grouping symbols. For example, A added to B and the result multiplied by C is represented as $(A + B) \cdot C$. *Contrast with*: **postfix notation; prefix notation.** [610.1]

informal testing. Testing conducted in accordance with test plans and procedures that have not been reviewed and approved by a customer, user, or designated level of management. *Contrast with:* **formal testing.** [610.12]

informatics. *See:* **information science.** [610.2]

information. The meaning that humans assign to data by means of known conventions that are applied to the data. *See also*: **formatted information; information traffic; narrative information.** [610.2a, 610.5]

information center (IC). (1) A user-oriented computer system that provides non-technical users direct access to data and software for information processing tasks such as report generation, data modeling and manipulation, and word processing. *Syn:* **information resource center; information**

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

technology center. *See also:* **decision support services.** [610.2]
(2) Support personnel for a computer system as in (1). [610.2]

information efficiency. The efficiency with which information is handled by an organization. *See also:* **information traffic.** [610.2]

information graphics. The use of a computer to produce low quality, low cost graphical output for peer group presentations. *Contrast with:* **presentation graphics.** *Syn:* **peer graphics.** [610.2]

information hiding. A software development technique in which each module's interfaces reveal as little as possible about the module's inner workings and other modules are prevented from using information about the module that is not in the module's interface specification. *See also:* **encapsulation.** [610.12]

information interchange. The process of sending and receiving data in such a manner that the information content or meaning associated with the data is not altered during the transmission. *See also:* **data interchange.** [610.5]

information overload. A condition resulting from presentation of too much data to be assimilated and acted upon without further organization. *See also:* **exception reporting.** [610.2]

information processing. *See:* **data processing.** [610.2]

information resource center (IRC). *See:* **information center.** [610.2]

information retrieval. The techniques used to recover information from an organized body of knowledge. *See also:* **information storage.** [610.5]

information science. A branch of technology concerned with the way in which data are processed and transmitted through digital equipment. *Syn:* **informatics.** [610.2]

information separator (IS). Any control character used to delimit like units of data in a hierarchical arrangement of data. The name of the separator does not necessarily indicate the units of data that it separates. *Syn:* **separating character.** [610.5]

information storage. The theory and techniques for the organization, storage, and searching of an organized body of knowledge. *Note:* Generally refers to a large body of data. *See also:* **information retrieval; information system.** [610.5]

information storage and retrieval. *See:* **information storage; information retrieval.** [610.5]

information storage and retrieval system. *See:* **information system.** [610.5]

information system. A mechanism used for acquiring, filing, storing, and retrieving an organized body of knowledge. *Syn:* **information storage and retrieval system.** *See also:* **information storage and retrieval.** [610.5]

information technology center. *See:* **information center.** [610.2]

information traffic. The flow of information through an organization. Typically included are origination; production, consolidation, and presentation; reproduction; recording and storage; and distribution. *Syn:* **document traffic; paper traffic.** *See also:* **document cycle; information efficiency office automation.** [610.2]

inherent error. *See:* **inherited error.** [610.1]

inherited error. (1) An error that is input to a given operation, either from a previous operation or from the initial condition of a variable. *Syn:* **inherent error.** *Contrast with:* **propagated error.** [610.1]
(2) An error carried forward from a previous step in a sequential process. [610.12]

initial condition. The values assumed by the variables in a system, model, or simulation at the beginning of some specified duration of time. *Contrast with:* **boundary condition; final condition.** [610.3]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

initialize. To set a variable, register, or other storage location to a starting value. *See also:* **clear; reset.** [610.12]

initial program load. *See:* **bootstrap.** [610.12]

initial program loader. A bootstrap loader used to load that part of an operating system needed to load the remainder of the operating system. [610.12]

initial state. The values assumed by the state variables of a system, component, or simulation at the beginning of some specified duration of time. *Contrast with:* **final state.** [610.3]

inline code. A sequence of computer instructions that is physically contiguous with the instructions that logically precede and follow it. [610.12]

inline recovery. Recovery performed by resuming a process at a point preceding the occurrence of a failure. *Contrast with:* **backward recovery; forward recovery.** [610.5]

inorder traversal. The process of traversing a binary tree in a recursive fashion as follows: the left subtree is traversed in order, then the root is visited, then the right subtree is traversed in order. *Contrast with:* **postorder traversal; preorder traversal.** *Syn:* **symmetric traversal.** *See also:* **converse inorder traversal.** [610.5]

input. (1) Pertaining to data received from an external source. [610.12]
(2) Pertaining to a device, process, or channel involved in receiving data from an external source. [610.12]
(3) To receive data from an external source. [610.12]
(4) To provide data from an external source.
(5) Loosely, input data. [610.12]
Contrast with: **output.** [610.12]

input assertion. A logical expression specifying one or more conditions that program inputs must satisfy in order to be valid. *Contrast with:* **loop assertion; output assertion.** *See also:* **inductive assertion method.** [610.12]

input buffer. *See:* **buffer.** [610.5]

input/output area. *See:* **buffer.** [610.5]

input-output coupling. *See:* **data coupling.** [610.12]

input/output model. *See:* **black box model.** [610.3]

input-process-output. A software design technique that consists of identifying the steps involved in each process to be performed and identifying the inputs to and outputs from each step. *Note:* A refinement called hierarchical input-process-output identifies the steps, inputs, and outputs at both general and detailed levels of detail. *See also:* **data structurecentered design; input-process-output chart; modular decomposition; objectoriented design; rapid prototyping; stepwise refinement; structured design; transaction analysis; transform analysis.** [610.12]

input-process-output (IPO) chart. A diagram of a software system or module, consisting of a rectangle on the left listing inputs, a rectangle in the center listing processing steps, a rectangle on the right listing outputs, and arrows connecting inputs to processing steps and processing steps to outputs. *See also:* **block diagram; box diagram; bubble chart; flowchart; graph; structure chart.** [610.12]

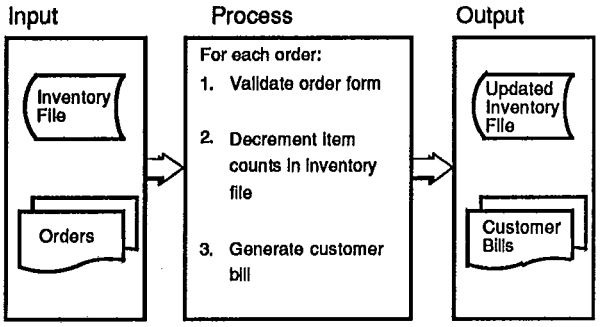


Fig 46
Input-Process-Output Chart

insertion character. A character within a picture specification that represents a character that is inserted into the representation only under certain circumstances; for example, the value 1234, when represented using the picture specification 9,999 (the comma is the insertion character), is "1,234." [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

insertion sort. A sort in which each item in the set to be sorted is inserted into its proper position among those items already considered. *Syn*: **straight insertion sort.** *See also*: **address calculation sort; binary insertion sort; distribution counting sort; diminishing increment sort; linear sort; list insertion sort; radix insertion sort; tree insertion sort; two-way insertion sort.** [610.5]

inspection. A static analysis technique that relies on visual examination of development products to detect errors, violations of development standards, and other problems. Types include code inspection; design inspection. [610.12]

installation and checkout phase. The period of time in the software life cycle during which a software product is integrated into its operational environment and tested in this environment to ensure that it performs as required. [610.12]

installation manual. A document that provides the information necessary to install a system or component, set initial parameters, and prepare the system or component for operational use. *See also:* **diagnostic manual; operator manual; programmer manual; support manual; user manual.** [610.12]

instantaneous access. *See*: **immediate access.** [610.5]

instantiation. The process of substituting specific data, instructions, or both into a generic program unit to make it usable in a computer program. [610.12]

instruction. *See:* **computer instruction.** [610.12]

instructional character. *See*: **control character.** [610.5]

instructional game. An instruction method employed by some computer-assisted instruction systems, in which a game is used to instruct the student on some subject. *Contrast with:* **question-and-answer interaction; simulation.** [610.2]

instructional simulation. A simulation intended to provide an opportunity for learning or to evaluate learning or educational potential; for example, a simulation in which a mock-up of an airplane cockpit is used to train student pilots. *Syn*: **academic simulation; tutorial simulation.** [610.3]

instruction counter. A register that indicates the location of the next computer instruction to be executed. *Syn:* **program counter.** [610.12]

instruction cycle. The process of fetching a computer instruction from memory and executing it. *See also:* **instruction time.** [610.12]

instruction format. The number and arrangement of fields in a computer instruction. *See also:* **address field; address format; operation field.** [610.12]

instruction length. The number of words, bytes, or bits needed to store a computer instruction. *See also:* **instruction format.** [610.12]

instruction modifier. A word or part of a word used to alter a computer instruction. [610.12]

instruction repertoire. *See:* **instruction set.** [610.12]

instruction set. The complete set of instructions recognized by a given computer or provided by a given programming language. *Syn:* **instruction repertoire.** [610.12]

instruction time. The time it takes a computer to fetch an instruction from memory and execute it. *See also:* **instruction cycle.** [610.12]

instrument. In software and system testing, to install or insert devices or instructions into hardware or software to monitor the operation of a system or component. [610.12]

instrumentation. Devices or instructions installed or inserted into hardware or software to monitor the operation of a system or component. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

integer. A positive or negative whole number, including zero. *Syn:* **integral number.** [610.1]

integer arithmetic. Fixed-point arithmetic in which the radix point is assumed to lie immediately to the right of the least significant digit in each numeral; that is, all numbers are assumed to be integers. [610.1]

integer data. Numeric data used to represent whole numbers; that is, numeric values without fractional parts. For example, 0, +1, -1, +2, -2, ... *See also*: **fixed-point data; packed decimal data; unsigned packed decimal data; zoned decimal data.** [610.5]

integer programming. In operations research, a class of procedures for locating the maximum or minimum of a function under given constraints, one of which is that some or all variables must have integer values. *Syn:* **discrete programming.** [610.2]

integer type. A data type whose members can assume only integer values and can be operated on only by integer arithmetic operations, such as addition, subtraction, and multiplication. *Contrast with:* **character type; enumeration type; logical type; real type.** [610.12]

integer variable. A variable that may assume only integer (non-fractional) values. [610.1]

integrated data dictionary. A data dictionary that is functionally involved in data accesses, performing required checks for value limits and data types and disallowing illegal modifications to data elements within the system that is described. [610.5]

integrated data processing (IDP). The use of computers to coordinate a number of processes and improve overall efficiency by reducing or eliminating redundant data entry or processing operations. [610.2]

integration. The process of combining software components, hardware components, or both into an overall system. [610.12]

integration testing. Testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them. *See also:* **component testing; interface testing; system testing; unit testing.** [610.12]

integrity. The degree to which a system or component prevents unauthorized access to, or modification of, computer programs or data. *See also*: **data integrity; database integrity.** [610.5, 610.12]

intelligent data model. A data model that describes the logic, controls, and constraints that should be applied whenever the data are accessed. [610.5]

interactive. Pertaining to a system or mode of operation in which each user entry causes a response from or action by the system. *Contrast with:* **batch.** *See also:* **conversational; on-line; real time.** [610.2a, 610.12]

interactive compiler. *See:* **incremental compiler.** [610.12]

interactive language. A nonprocedural language in which a program is created as a result of interactive dialog between the user and the computer system. The system provides questions, forms, and so on, to aid the user in expressing the results to be achieved. *See also:* **declarative language; rule-based language.** [610.12]

interactive plotting. The use of a display device to view the output of a graphic or computational process. Applications include computer-assisted instruction, computer-aided design, and control operations. [610.2]

interblock gap. An area between two consecutive blocks. *Syn*: **block gap; record gap.** [610.5]

intercharacter spacing. In text formatting, the amount of space left between characters on a line. *Contrast with:* **interword spacing.** *See also:* **incremental justification; kerning.** *Syn:* **letter spacing.** [610.2]

interface. 1) A shared boundary across which information is passed. [610.12]
(2) A hardware or software component that connects two or more other components for

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

the purpose of passing information from one to the other. [610.12]
(3) To connect two or more components for the purpose of passing information from one to the other. [610.12]
(4) To serve as a connecting or connected component as in (2). [610.12]

interface control. (1) In configuration management, the process of: (a) identifying all functional and physical characteristics relevant to the interfacing of two or more configuration items provided by one or more organizations, and (b) ensuring that proposed changes to these characteristics are evaluated and approved prior to implementation. [610.12]
(2) (DoD usage) In configuration management, the administrative and technical procedures and documentation necessary to identify functional and physical characteristics between and within configuration items provided by different developers, and to resolve problems concerning the specified interfaces. *See also:* **configuration control.** [610.12]

interface requirement. A requirement that specifies an external item with which a system or system component must interact, or that sets forth constraints on formats, timing, or other factors caused by such an interaction. *Contrast with:* **design requirement; functional requirement; implementation requirement; performance requirement; physical requirement.** [610.12]

interface specification. A document that specifies the interface characteristics of an existing or planned system or component. [610.12]

interface testing. Testing conducted to evaluate whether systems or components pass data and control correctly to one another. *See also:* **component testing; integration testing; system testing; unit testing.** [610.12]

interframe coding. An image compression technique in which a sequence of images is compressed by taking advantage of redundancies between successive images. [610.4]

interior. The set of pixels in a region of a digital image that are not adjacent to pixels in the region's complement. *Contrast with:* **border.** [610.4]

interleave. To alternate the elements of one sequence with the elements of one or more other sequences so that each sequence retains its identity; for example, to alternately perform the steps of two different tasks in order to achieve concurrent operation of the tasks. [610.12]

interleaved array. In PL/1, an array whose name refers to non-contiguous storage. [610.5]

intermediate product. *See:* **partial product.** [610.1]

intermittent fault. A temporary or unpredictable fault in a component. *See also:* **random failure; transient error.** [610.12]

internal label. A machine-readable label recorded on a data medium that provides information about the data recorded on the medium. *Contrast with*: **external label.** *See also*: **end-of-volume label; end of file; header label.** [610.5]

internal merge sort. A merge sort performed within main storage. *See also*: **Batcher's parallel sort; list merge sort; two-way merge sort.** *Contrast with*: **external merge sort.** [610.5]

internal model. A data model depicting entities within the conceptual schema of a database for a specific application. *Contrast with*: **external database model.** [610.5]

internal node. *See*: **nonterminal node.** [610.5]

internal record. A record within an internal view. *Syn*: **stored record.** [610.5]

internal schema. (1) A description of the format and layout of the entire contents of a database including the data as well as overhead portions such as indices. *Note*: Written using data definition language. *Contrast with*: **conceptual schema; external schema.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

(2) A description of the data as it is physically stored in a database, including a description of the environment in which the database is to reside. [610.5]

internal sort. A sort performed within main storage. *Syn*: **in-core sort.** *Contrast with*: **external sort.** [610.5]

internal variable. *See:* **endogenous variable.** [610.3]

internal view. The format, layout, and contents of the entire data content and overhead content of a database, as described in an internal schema. *Note*: There may be many external views of a database, but only one internal view. [610.5]

interoperability. The ability of two or more systems or components to exchange information and to use the information that has been exchanged. *See also:* **compatibility.** [610.12]

interpolation search. A searching technique in which, at each step of the search, an estimate is made of where the desired record is apt to be. *Syn*: **estimated entry search; external entry search.** *Contrast with*: **binary search; dichotomizing search; fibonacci search.** [610.5]

interpret. To translate and execute each statement or construct of a computer program before translating and executing the next. *Contrast with:* **assemble; compile.** [610.12]

interpreter. A computer program that translates and executes each statement or construct of a computer program before translating and executing the next. *Contrast with:* **assembler; compiler.** [610.12]

interpretive code. Computer instructions and data definitions expressed in a form that can be recognized and processed by an interpreter. *Contrast with:* **assembly code; compiler code; machine code.** [610.12]

interrupt. (1) The suspension of a process to handle an event external to the process. *Syn:* **interruption.** *See also:* **interrupt latency; interrupt mask; interrupt priority; interrupt service routine; priority interrupt.** [610.12]
(2) To cause the suspension of a process. [610.12]
(3) Loosely, an interrupt request. [610.12]

interruption. *See:* **interrupt.** [610.12]

interrupt latency. The delay between a computer system's receipt of an interrupt request and its handling of the request. *See also:* **interrupt priority.** [610.12]

interrupt mask. A mask used to enable or disable interrupts by retaining or suppressing bits that represent interrupt requests. [610.12]

interrupt priority. The importance assigned to a given interrupt request. This importance determines whether the request will cause suspension of the current process and, if there are several outstanding interrupt requests, which will be handled first.[610.12]

interrupt request. A signal or other input requesting that the currently executing process be suspended to permit performance of another process. [610.12]

interrupt service routine. A routine that responds to interrupt requests by storing the contents of critical registers, performing the processing required by the interrupt request, restoring the register contents, and restarting the interrupted process. [610.12]

intersection. (1) A relational operator that combines two relations having the same degree and results in a relation containing all of the tuples that are in both of the original relations. *See also*: **difference; join; product; projection; selection; union.** [610.5]
(2) *See:* **AND.** [610.1]

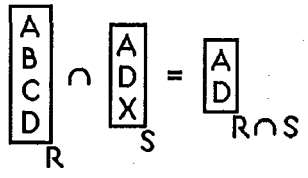


Fig 47
Intersection

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

interval-oriented simulation. A continuous simulation in which simulated time is advanced in increments of a size suitable to make implementation possible on a digital system. [610.3]

interword spacing. In text formatting, the amount of space left between words on a line. *Contrast with:* **intercharacter spacing.** *See also:* **line filling.** [610.2]

invalid character. * *See*: **illegal character.** * Deprecated. [610.5]

invariant. An assertion that should always be true for a specified segment or at a specified point of a computer program. [610.12]

inverse. *See:* **ones complement.** [610.1]

inverse binary state. *See:* **ones complement.** [610.1]

inversion. (1) In Boolean algebra, the same as NOT. [610.1]
(2) The process of taking the reciprocal of a number. [610.1]
(3) The process of constructing an inverted list to be used to access a set of records. [610.5]

invert. (1) To change a binary variable to its opposite logic state. [610.1]
(2) To take the reciprocal of a number. [610.1]

inverted file. (1) A file whose elements may be retrieved by searching either the primary key or secondary key of each record. *Note*: An inverted file is distinguished from other files by the logical relationship and organization of items and records. In an inverted file, each value of each data item in the records appears exactly once, instead of once in each record. *See also*: **fully inverted file; partially inverted file; secondary index.** [610.5]
(2) A file whose initial sequence has been reversed or whose contents may be searched in reverse order. [610.5]

inverted list. (1) A list whose contents may be retrieved by searching either the primary key or the secondary key of each element. [610.5]
(2) A technique for organizing records in which the primary keys for records that have equivalent values for a given secondary key are stored in a secondary index. *Contrast with*: **multilist.** *See also*: **inversion.** [610.5]

IPO chart. Acronym for **input-process-output chart.** [610.12]

IPSE. Acronym for **integrated programming support environment.** *See:* **programming support environment.** [610.12]

IRC. Acronym for **information resource center.** *See:* **information center.** [610.2]

irrational number. A real number that is not a rational number. *Contrast with*: **rational number.** [610.5]

IS. Acronym for **information separator.** [610.5]

ISAM. Acronym for **indexed sequential access method.** [610.5]

ISP. Acronym for **information structure perspective.** [610.5]

item. One member of a group; for example, a field in a record or a record in a file. *See also*: **data item.** [610.5]

item condition. A disjunction of two or more atomic conditions such that the name of the data item is the same in each atomic condition. For example, "LASTNAME = 'JONES' or LASTNAME = 'SMITH' or LASTNAME = 'GREEN.'" *Note*: The disjunction may be implied, as in the example "LASTNAME = ('JONES','SMITH','GREEN'). *See also*: **query condition; record condition.** [610.5]

iteration. (1) The process of performing a sequence of steps repeatedly. *See also:* **loop; recursion.** [610.12]
(2) A single execution of the sequence of steps in (1). [610.12]

iterative construct. *See:* **loop.** [610.12]

iterative document. A document that will be produced multiple times with relatively few changes in the text. For example, a letter meant to be prepared 100 times, each with a

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052856 0

different name and address in the salutation. *Syn:* **form letter.** [610.2]

IV&V. Acronym for **independent verification and validation.** [610.12]

JCL. Acronym for **job control language.** [610.12]

job. A user-defined unit of work that is to be accomplished by a computer. For example, the compilation, loading, and execution of a computer program. *See also:* **job control language; job step; job stream.** [610.12]

job control language (JCL). A language used to identify a sequence of jobs, describe their requirements to an operating system, and control their execution. [610.12]

job function. A group of engineering processes that is identified as a unit for the purposes of work organization, assignment, or evaluation. Examples are design, testing, or configuration management. [610.12]

job step. A user-defined portion of a job, explicitly identified by a job control statement. A job consists of one or more job steps. [610.12]

job stream. A sequence of programs or jobs set up so that a computer can proceed from one to the next without the need for operator intervention. *Syn:* **run stream.** [610.12]

join. A relational operator that combines two relations having a common attribute and which results in a relation containing all of the attributes from both of the original relations. *See also:* **intersection; difference; product; projection; selection; union.** [610.5]

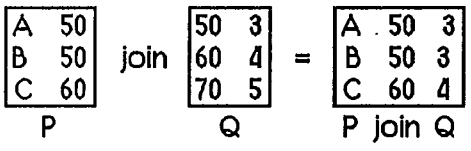


Fig 48
Join

join dependency. A type of dependency within a relation R, in which R is join dependent on $X, Y, ..., Z$ (subsets of attributes in R) if and only if R is equal to the join of its projections on $X, Y, ..., Z$. *See also:* **functional dependency.** [610.5]

journal. A chronological record of the changes made to a set of data. *Note:* This record may be used as an audit trail to reconstruct a previous version of the data. *Syn:* **log.** [610.2, 610.5]

jump. (1) To depart from the implicit or declared order in which computer program statements are being executed. *Syn:* **transfer.** [610.12]
(2) A program statement that causes a departure as in (1). *Contrast with:* **case; if-then-else.** *See also:* **branch; go to.** [610.12]
(3) The departure described in (1). *See also:* **conditional jump; unconditional jump.** [610.12]

justification. (1) In text formatting, the process of aligning text to form even margins or to achieve desired vertical spacing. *See also:* **full justification; hyphenless justification; incremental justification; left justification; right justification; vertical justification.** [610.2]
(2) The result of the process in (1). [610.2]

justification range. In text formatting, the permitted minimum and maximum space that can be inserted between words or characters by a justification routine. [610.2]

justification routine. In text formatting, a routine that produces justified text by using interword or intercharacter spacing. [610.2]

justify. To shift a numeral so that the most significant digit, the least significant digit, or the radix point is placed at a specific position in a register. [610.1]

k. Abbreviation for **kilo (1).** [610.1a]

K. Abbreviation for **kilo (2).** [610.1a]

Karnaugh map. A rectangular diagram of a logical expression drawn with overlapping rectangles representing a unique combination of the logic variables such that an intersection is shown for all combinations. The rows and columns are headed with combi-

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

nations of the variables in a Gray code sequence. *See also:* **logic map; Mahoney map.** [610.1]

kernel. (1) That portion of an operating system that is kept in main memory at all times. *Syn:* **nucleus; resident control program.** [610.12]
(2) A software module that encapsulates an elementary function or functions of a system. *See also:* **security kernel.** [610.12]

kerning. In text formatting, the use of inter-character spacing to expand or compact a word or a line of text. *Syn:* **mortising; white space reduction; white space expansion.** [610.2]

key. (1) In data management, a data element or concatenation of data elements that identifies an item within a set of items. *Note*: Such a data element is also known as a key field. *See also*: **concatenated key; key value; primary key; secondary key; sort key.** *Syn*: **key field; sequence field.** [610.5]
(2) In a relational data model, one or more attributes that, when taken together, identify the relation to which the attributes belong. [610.5]
(3) In a tree, the portion of each node that identifies that node. [610.5]

key compression. The elimination of data from the beginning and the end of a key in which these characters are not needed to distinguish the key from other keys in the set. [610.5]

key field. *See*: **key.** [610.5]

keyed access. *See*: **indexed access.** [610.5]

key folding function. A hash function in which the original key is split into two or more parts and some portion of their sum is returned as the hash value. For example, in the function below, the key is divided into three parts and the sum of the three parts is returned as the hash value. [610.5]

Original key	Calculation	Hash value
96472135	964 + 721 + 35 = 1738	1738
90007810	900 + 078 + 10 = 988	988

key generation. The process of generating the key values for the items in a set according to some algorithm. [610.5]

key letter in context (KLIC) index. A variation of a keyword in context (KWIC) index in which letters are used as the fundamental indexing units instead of keywords. *See also:* **key phrase in context index.** [610.2]

key phrase in context (KPIC) index. A variation of a keyword in context (KWIC) index in which phrases are used as the fundamental indexing units instead of keywords. *See also:* **key letter in context index.** [610.2]

key range. A particular range of values of the keys found in some set of data. *Note*: Key ranges may be used to partition the set into subsets. [610.5]

key sequence. Pertaining to a set of data that has been sequenced according to the value of some key. [610.5]

key sorting. A sorting technique in which a table of sort keys and corresponding addresses that point to the items to be stored are manipulated instead of moving the items themselves. *See also*: **address table sorting.** [610.5]

key transformation. In searching, the process of mapping a set of keys into a set of integers, using a hash function. [610.5]

key transformation function. *See*: **hash function.** [610.5]

key value. The contents of a key. [610.5]

keyword. In automatic indexing, a significant word in a title or document that characterizes the content of the document. *Syn:* **descriptor; lead term.** [610.2]

keyword and context (KWAC) index. A type of keyword in context index in which items are presented in the form of a permutation index. *See also:* **permutation on subject headings index.** [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

keyword in context (KWIC) index. An automatic index in which keywords are placed in alphabetical order in a central column and the remainder of the information is given to the right and left, preserving the context. *See also:* **author and keyword in context index; key letter in context index; key-phrase in context index; keyword and context index.** *Contrast with:* **keyword out of context index.** [610.2]

keyword out of context (KWOC) index. An automatic index in which the keywords are extracted from their normal context and are displayed in a left hand column with the full context following on the right in its normal order. *See also:* **keyword out of title index; word and author index.** *Contrast with:* **keyword in context index.** [610.2]

keyword out of title (KWOT) index. A type of keyword out of context index in which the items being indexed are document titles. [610.2]

kilo (K or k). (1) A prefix indicating one thousand (10^3). [610.1a]
(2) In statements involving size of computer storage, a prefix indicating 2^{10}, or 1024. [610.1]

KLIC. Acronym for **key letter in context.** *See:* **key letter in context index.** [610.2]

knowledge base. A collection of interrelated information, facts, or statements. [610.5, 610.12]

KOPS. Acronym for kilo-operations per second; that is, thousands of operations per second. A measure of computer processing speed. *See also:* **MFLOPS; MIPS.** [610.12]

Kordic algorithm. A widely used algorithm that calculates the sine and cosine of an angle using only addition, subtraction, and shifting operations in scaled arithmetic. [610.1]

KPIC. Acronym for **key phrase in context.** *See:* **key phrase in context index.** [610.2]

KWAC. Acronym for **keyword and context.** *See:* **keyword and context index.** [610.2]

KWIC. Acronym for **keyword in context.** *See:* **in context index.** [610.2]

KWOC. Acronym for **keyword out of context.** *See:* **keyword out of context index.** [610.2]

KWOT. Acronym for **keyword out of title.** *See:* **keyword out of title index.** [610.2]

label. (1) A name or identifier assigned to a computer program statement to enable other statements to refer to that statement. [610.12]
(2) One or more characters within or attached to a set of data, that identify or describe the data. *Syn:* **identifier.** [610.5a, 610.12]

laboratory simulation. A simulation developed and used under highly controlled conditions; for example, a simulation of a medical technique implemented in the controlled environment of a laboratory. [610.3]

lagged variable. *See:* **lag variable.** [610.3]

lag variable. (1) In a discrete simulation, a variable that is an output of one period and an input for some future period. [610.3]
(2) In an analog simulation, a variable that is a function of an output variable and that is used as input to the simulation to provide a time delay response or feedback. *Syn:* **lagged variable; serially-correlated variable.** [610.3]

landscape image. *See:* **comic-strip oriented image.** [610.2]

language. (1) A systematic means of communicating ideas by the use of conventionalized signs, sounds, gestures, or marks and rules for the formation of admissible expressions. [610.12]
(2) A means of communication, with syntax and semantics, consisting of a set of representations, conventions, and associated rules used to convey information. [610.12]
See also: **computer language.** [610.12]

language processor. A computer program that translates, interprets, or performs other tasks required to process statements expressed in a given language. *See also:* **assembler; compiler; interpreter; translator.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

language standard. A standard that describes the characteristics of a language used to describe a requirements specification, a design, or test data. [610.12]

laptop computer. A portable computer designed for use on one's lap. [610.2]

last-in, first-out (LIFO). (1) A technique for managing a set of items to which additions and deletions are to be made; items are appended to one end and retrieved from that same end. *See also*: **stack.** [610.5]
(2) Pertaining to a system in which the next item to exit the system is the item that has been in the system for the shortest time. *Contrast with*: **first-in, first-out.** *Syn*: **first-in, last-out.** [610.5]

latency. The time interval between the instant at which an instruction control unit issues a call for data and the instant at which the transfer of data is started. [610.12]

lateral compression. In software design, a form of demodularization in which two or more modules that execute one after the other are combined into a single module. *Contrast with:* **downward compression; upward compression.** [610.12]

layout character. *See*: **format effector character.** [610.5]

LC-MARC (or LCMARC). Acronym for **Library of Congress MAchine-Readable Cataloging.** *See:* **Machine Readable Cataloging.** [610.2]

lead term. *See:* **keyword.** [610.2]

lead variable. (1) In a discrete simulation, a variable that is an output of one period and that predicts what the output of some future period will be. [610.3]
(2) In an analog simulation, a variable that is a function of an output variable and that is used as input to the simulation to provide advanced time response or feedback. [610.3]

leader. (1) In text formatting, a character used to lead the reader's eye across the page and to indicate logical connection between two items on a line. For example, one of a series of periods used in a table of contents to associate a chapter title with its corresponding page number. [610.2]
(2) The blank section of magnetic tape at the beginning of a reel. [610.5]

leading. In photocomposition, the use of white space between lines and paragraphs of a document. *See also:* **reverse leading.** [610.2]

leading decision. A loop control that is executed before the loop body. *Contrast with:* **trailing decision.** *See also:* **WHILE.** [610.12]

leading zero. A zero that precedes the first non-zero digit in a numeric representation; for example, the two zeros in "00324.6." *Contrast with*: **trailing zero.** [610.5]

leaf. (1) A terminal node in a search tree. [610.5]
(2) In a tree, a node that has no children. [610.5]

least significant bit (LSB). The bit having the smallest effect on the value of a binary numeral; usually the rightmost bit. [610.1]

least significant character. The character in the rightmost position in a character string. *Contrast with*: **most significant character.** [610.1, 610.5]

least significant digit. The digit having the smallest effect on the value of a numeral; usually the right-most digit; for example, the 4 in 756.4. *Contrast with*: **most significant digit.** [610.1a, 610.5]

left justification. In text formatting, justification of text such that the left margin is aligned. *Contrast with:* **ragged left margin.** [610.2]

left-threaded tree. A tree in which the left link field in each terminal node is made to point to its predecessor with respect to a particular order of traversal. *Contrast with*: **right-threaded tree.** [610.5]

length. *See*: **block length; record length.** [610.5]

letter spacing. *See:* **intercharacter spacing.** [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

library. *See*: **data library; software library.** [610.5, 610.12]

library automation. The application of automated techniques to library operations such as processing of documents, interlibrary communication, and on-line catalog access. *See also:* **MAchine Readable Cataloging.** [610.2]

licensing standard. A standard that describes the characteristics of an authorization given by an official or a legal authority to an individual or organization to do or own a specific thing. [610.12]

life cycle. *See:* **software life cycle; system life cycle.** [610.12]

LIFO. Acronym for **last-in, first-out.** [610.5]

limit check. A consistency check that ensures that a certain item limit is not exceeded. For example, if a record can hold four transactions, a limit check will reveal an error situation if an attempt is made to add a fifth transaction to a record. [610.5]

limited-domain data element. A data element whose domain is bounded. For example, a data element SEX with a domain of [M,F]. [610.5]

linear array. A one-dimensional array.[610.5]

linear data structure. A nonprimitive data structure that can represent data that is one-dimensional in nature. For example, a vector. *Contrast with*: **nonlinear data structure.** [610.5]

linear linked list. *See*: **linked linear list.** [610.5]

linear list. A list that preserves the relationship of adjacency between data items in the list. [610.5]

linear optimization. *See:* **linear programming.** [610.2]

linear probing. Open-address hashing in which collision resolution is handled by inserting an item that has a duplicate hash value into the next available position in the hash table. *Syn*: **consecutive spill method.** *Contrast with*: **quadratic probing; random probing; uniform probing.** [610.5]

linear programming (LP). In operations research, a procedure for locating the maximum or minimum of a linear function of variables that are subject to linear constraints. *Contrast with:* **nonlinear programming.** [610.1a, 610.2]

linear quantizing. *See:* **equal interval quantizing.** [610.4]

linear search. *See*: **sequential search.** [610.5]

linear sort. (1) A insertion sort in which each item in the set to be sorted is inserted into the sorted set by scanning the sorted set sequentially to locate the proper place. [610.5] (2) A sort in which the items in the set to be sorted exist in a linear list. *Syn*: **straight line sort.** [610.5]

line-deletion character. A character within a line of terminal input specifying that it and all previous characters on the line are to be removed from the line; for example, if "*" is the line-deletion character in the string "ABCD*APPLE," the following would appear on the terminal: "APPLE." *See also*: **character-deletion character.** [610.5]

line detection. An image segmentation technique in which line pixels are identified by examining their neighborhoods. [610.4]

line editing. A method of text editing that allows the user to change text, with cursor control, on only one line at a time. Multiple lines may be viewed or changed through editing commands. *Syn:* **line-oriented editing.** *Contrast with:* **full-screen editing.** *See also:* **context editing.** [610.2]

line editor. A text editor that allows the user to change text, with cursor control, on only one line at a time. Multiple lines may be viewed or changed through editing commands. *Contrast with:* **full-screen editor.** [610.2]

line-ending zone. *See:* **hot zone.** [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

line-end zone. *See:* **hot zone.** [610.2]

line filling. In text formatting, the use of extra interword spacing to form even margins. *Contrast with:* **incremental justification.** [610.2]

line-oriented editing. *See:* **line editing.** [610.2]

line pixel. A pixel contained in an arc that approximates a straight line. [610.4]

link. (1) *See*: **pointer.** [610.5]
(2) To establish a pointer; for example, to link two items in a hierarchy. [610.5]
(3) In relation theory, a relationship between two or more entities or records. [610.5]
(4) To append an item to a linked list. *See also*: **link field; push.** [610.5]
(5) To create a load module from two or more independently translated object modules or load modules by resolving cross-references among them. *See also:* **linkage editor.** [610.12]
(6) A part of a computer program, often a single instruction or address, that passes control and parameters between separate modules of the program. *Syn:* **linkage.** [610.12]
(7) To provide a link as in (6). [610.12]

linkage. *See:* **link (6).** [610.12]

linkage editor. A computer program that creates a single load module from two or more independently translated object modules or load modules by resolving cross-references among the modules and, possibly, by relocating elements. May be part of a loader. *Syn:* **linker.** *See also:* **linking loader.** [610.12]

linked linear list. A linear list in which each item contains a pointer to the next item in the list, making it unnecessary for the items to be physically sequential. *Note*: the items are still logically adjacent. *Syn*: **linear linked list.** [610.5]

linked list. A list in which each item contains a pointer to the next or preceding item in the list, making it unnecessary for the items to be physically sequential. *Note*: Unless the list is circular, the last item in the list contains a null link field. *Syn*: **chain; chained list; one-way chain; singly linked list.** *See also*: **circularly linked list; doubly linked list; linked linear list.** [610.5]

linker. *See:* **linkage editor.** [610.12]

link field. (1) A field in each item of a linked list, containing a pointer to the next or preceding item in the list. *Syn*: **chain field.** [610.5]
(2) In a tree, that portion of each node that contains a pointer to other nodes in the tree. [610.5]

linking loader. A computer program that reads one or more object modules into main memory in preparation for execution, creates a single load module by resolving cross-references among the separate modules, and, in some cases, adjusts the addresses to reflect the storage locations into which the code has been loaded. *See also:* **absolute loader; relocating loader; linkage editor.** [610.12]

list. (1) A set of data items, each of which has the same data definition. *See also*: **linear list; linked list; ordered list; queue; stack; unordered list.** [610.5, 610.12]
(2) To print or otherwise display a set of data items. [610.5, 610.12]
(3) * *See*: **linked list.** [610.5]
* Deprecated.

listing. An ordered display or printout of data items, program statements, or other information. [610.12]

list insertion sort. An insertion sort implemented using the list sorting technique. For example, a linear sort. [610.5]

list merge sort. A merge sort implemented using the list sorting technique. [610.5]

list processing. The manipulation of data that is or is going to be stored in list structures. [610.5]

list processing language. A programming language designed to facilitate the manipulation of data expressed in the form of lists. Examples are LISP and IPL. *See also:* **algebraic language; algorithmic language; logic programming language.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052862 6

list sorting. A sorting technique in which the items to be sorted form a linked list and the links between the items in the list are manipulated in such a way that, in the final list, the items form a linked list in sorted order. *See also*: **address table sorting; key sorting.** [610.5]

list structure. (1) A list, each item of which is either a single data item or a list structure itself. [610.5]
(2) A data structure that contains one or more lists. *Syn*: **compound list.** [610.5]

literal. (1) Composed of characters, as in a literal variable name used to contain a customer's name. [610.5]
(2) In a source program, an explicit representation of the value of an item; for example, the word FAIL in the instruction: If $x = 0$ then print "FAIL". *See also:* **immediate data; figurative constant.**[610.12]

load. (1) To read machine code into main memory in preparation for execution and, in some cases, to perform address adjustment and linking of modules. *See also:* **loader.** [610.12]
(2) To copy computer instructions or data from external storage to internal storage or from internal storage to registers. *Contrast with:* **store (2).** *See also:* **fetch; move.** [610.12]
(3) To insert data values into a database that previously contained no data. *Syn*: **populate.** *See also*: **download; upload.** [610.5]

load-and-go. An operating technique in which there are no stops between the loading and execution phases of a computer program. [610.12]

loaded origin. The address of the initial storage location of a computer program at the time the program is loaded into main memory. *Contrast with:* **assembled origin.** *See also:* **offset (1); starting address.** [610.12]

loader. (1) A computer program that reads machine code into main memory in preparation for execution and, in some cases, adjusts the addresses and links the modules. Types include absolute loader, linking loader, relocating loader. *See also:* **bootstrap; linkage editor.** [610.12]
(2) Any program that reads programs or data into main memory. [610.12]

loading factor. (1) The maximum amount of usable space in a physical block after accounting for block overhead. [610.5]
(2) The ratio of the number of stored entities in a file to the maximum number of entries that can be stored in a unit of data medium. [610.5]

load map. A computer-generated list that identifies the location or size of all or selected parts of memory-resident code or data. [610.12]

load module. A computer program or subprogram in a form suitable for loading into main storage for execution by a computer; usually the output of a linkage editor. *See also:* **object module.** [610.12]

local compaction. In microprogramming, compaction in which microoperations are not moved beyond the boundaries of the single entry, single exit sequential blocks in which they occur. *Contrast with:* **global compaction.** [610.12]

local data. Data that can be accessed by only one module or set of nested modules in a computer program. *Contrast with:* **global data.** [610.12]

local variable. A variable that can be accessed by only one module or set of nested modules in a computer program. *Contrast with:* **global variable.** [610.12]

lock. (1) To exclude users from updating data that is being updated by another user. *Note*: Depending on the implementation, locking may occur on a field, record or an entire file. *See also*: **deadlock; exclusive lock.** [610.5]
(2) To exclude users from accessing data. *Syn*: **field-locking; file-locking; record-locking.** [610.5]

locking. In code extension characters, having the characteristic that a change in interpretation applies to all coded representations following, or to all coded representations of a given class, until the next appropriate code

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

extension character occurs. *Contrast with*: **nonlocking.** [610.5]

locking shift character. A shift-out character that causes all characters that follow to be interpreted as members of a different character set from the original one until the shift-in character of the original character set is encountered. *Contrast with*: **nonlocking shift character.** [610.5]

lockout. A computer resource allocation technique in which shared resources (especially data) are protected by permitting access by only one device or process at a time. *See also:* **deadlock; semaphore.** [610.12]

log. *See:* **journal.** [610.2, 610.5]

logarithmic search. *See*: **binary search.** [610.5]

logic add. *See:* **OR.** [610.1]

logical. (1) Pertaining to a view or description of data that does not depend on the characteristics of the computer system or the physical storage. [610.5]
(2) Pertaining to the form of data organization, hardware or system that is processed by an application program; it may be different from the real (physical) form. *Contrast with*: **physical.** [610.5]

logical add. *See:* **OR.** [610.1]

logical child segment. (1) In a hierarchical database, a child segment in a logical database. [610.5]
(2) A pointer segment that establishes a child/parent relationship between a physical segment and a logical parent segment. *See also*: **physical child segment.** [610.5]

logical cohesion. A type of cohesion in which the tasks performed by a software module perform logically similar functions; for example, processing of different types of input data. *Contrast with:* **coincidental cohesion; communicational cohesion; functional cohesion; procedural cohesion; sequential cohesion; temporal cohesion.** [610.12]

logical comparison. The examination of two binary variables to determine whether they have the same value. *Syn:* **logic comparison; logic decision; logical decision.** [610.1]

logical connective. *See:* **logical operator.** [610.1]

logical data model. A data model that represents the meaning of the data contained in a data structure. *Contrast with*: **physical data model.** [610.5]

logical data. Data used to represent the result of some logical operation. [610.5]

logical database. (1) A database as it is perceived by its users. *Syn*: **application view; logical view.** [610.5]
(2) A database containing a collection of related segments that may reside in one or more physical databases. *Note*: A logical database is sometimes referred to as a logical view or application view of a physical database. *See also*: **logical segment; view integration.** [610.5]
(3) A database containing a subset of the segments in a physical database. *Note*: The root segment in the logical database must be the root segment in the physical database. *Contrast with*: **physical database.** [610.5]

logical decision. *See:* **logical comparison.** [610.1]

logical diagram. *See:* **logic diagram.** [610.1]

logical difference. A set consisting of all elements belonging to set A but not to set B, when two sets of elements, A and B, are given. *Syn:* **logic difference.** [610.1]

logical expression. A combination of symbols and variables representing a logical relationship. [610.1]

logical file. A file independent of its physical environment. Portions of the same logical file may be located in different physical files, and several logical files or parts of logical files may be located in one physical file. [610.5]

logical multiply. *See:* **AND.** [610.1]

logical operation. (1) An operation involving logical variables and operators. [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

(2) Loosely, any nonarithmetic computer operation. *Syn:* **logic operation.** [610.1]

logical operator. A symbol that represents a logical operation to be performed on the associated operands. *Syn:* **logic operator; logical connective.** [610.1]

logical parent segment. (1) In a hierarchical database, a parent segment in a logical database. *See also*: **physical parent segment.** [610.5]
(2) A segment that is pointed to by a logical child segment, establishing a parent/child relationship between the logical parent segment and some physical segment. *Note*: A logical parent segment may also be a physical parent segment. [610.5]

logical product. The result obtained from the AND operation. *Syn:* **logic product.** [610.1]

logical record. A record independent of its physical environment. *Note*: Portions of the same logical record may be located in different physical records, and several logical records or parts of logical records may be located in one physical record.[610.5]

logical schema. A schema that defines a data model. [610.5]

logical segment. A segment in a logical database. *See also*: **logical child; logical parent; logical twin.** [610.5]

logical shift. A shift that affects all positions in a register, word, or numeral, including the sign position. For example, +231.702 shifted two places to the left becomes 23170.200. *Note:* A logical shift may be applied to the multiple-precision representation of a number. *Syn:* **logic shift; nonarithmetic shift.** *Contrast with:* **arithmetic shift.** [610.1]

logical structure. *See*: **data structure.** [610.5]

logical sum. The result obtained from the OR operation. *Syn:* **logic sum.** [610.1]

logical symbol. *See:* **logic symbol.** [610.1]

logical trace. An execution trace that records only branch or jump instructions. *See also:* **execution trace; retrospective trace; subroutine trace; symbolic trace; variable trace.** [610.12]

logical twin segment. A twin segment in a logical database. *Contrast with*: **physical twin segment.** [610.5]

logical type. A data type whose members can assume only logical values (usually TRUE and FALSE) and can be operated on only by logical operators, such as AND, OR, and NOT. *Contrast with:* **character type; enumeration type; integer type; real type.** [610.12]

logical variable. *See:* **switching variable.** [610.1]

logical view. *See*: **external schema; logical database.** [610.5]

logically deleted. Pertaining to a record that no longer appears available to the user but is physically present in the file. *See also:* **active; inactive; purged.** [610.2]

logic comparison. *See:* **logical comparison.** [610.1]

logic decision. *See:* **logical comparison.**[610.1]

logic diagram. A graphical representation of a system's logic elements and their interconnections. *Syn:* **logical diagram.** [610.1]

logic difference. *See:* **logical difference.**[610.1]

logic function.* *See:* **switching function.**
*Deprecated. [610.1]

logic map. A worksheet used by logic designers in the process of logic development, simplification, or optimization. *See also:* **Karnaugh map; Mahoney map.** [610.1]

logic multiply. *See:* **AND.** [610.1]

logic operation. *See:* **logical operation.** [610.1]

logic operator. *See:* **logical operator.** [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

logic product. *See:* **logical product.** [610.1]

logic programming language. A programming language used to express programs in terms of control constructs and a restricted predicate calculus; for example, PROLOG. *See also:* **algebraic language; algorithmic language; list processing language.** [610.12]

logic shift. *See:* **logical shift.** [610.1]

logic state. One of the two possible values a binary variable may assume. [610.1]

logic sum. *See:* **logical sum.** [610.1]

logic symbol. (1) A symbol used to denote a logical operator. [610.1]
(2) A symbol used to graphically represent a logic element. *Syn:* **logical symbol.** [610.1]

logic variable.* *See:* **switching variable.**
*Deprecated. [610.1]

look up. To use a code-decode table or look-up table to obtain data values or other information. *See also:* **table look-up.** [610.5a]

look-up table. A table of values used in obtaining the value of a function using a table look-up procedure. *See also:* **code-decode table.** [610.1, 610.5a]

loop. (1) A sequence of computer program statements that is executed repeatedly until a given condition is met or while a given condition is true. *Syn:* **iterative construct.** *See also:* **loop body; loop control; UNTIL; WHILE.** [610.12]
(2) To execute a sequence of computer program statements as in (1). [610.12]

loop assertion. A logical expression specifying one or more conditions that must be met each time a particular point in a program loop is executed. *Syn:* **loop invariant.** *Contrast with:* **input assertion; output assertion.** *See also:* **inductive assertion method.** [610.12]

loopback testing. Testing in which signals or data from a test device are input to a system or component, and results are returned to the test device for measurement or comparison. [610.12]

loop body. The part of a loop that accomplishes the loop's primary purpose. *Contrast with:* **loop control.** [610.12]

loop control. The part of a loop that determines whether to exit from the loop. *Contrast with:* **loop body.** *See also:* **leading decision; trailing decision.** [610.12]

loop-control variable. A program variable used to determine whether to exit from a loop. [610.12]

loop invariant. *See:* **loop assertion.** [610.12]

lossless encoding. Any image compression technique that represents gray levels compactly but permits exact reconstruction of the image; for example, contour encoding; run length encoding. [610.4]

low level language. *See:* **assembly language.** [610.12]

low-order. Pertaining to the right-most digit or digits of a numeral. [610.1]

low-order position. The rightmost position in a string; for example, the letter "E" in "APPLE" or the "5" in "965." *Contrast with:* **high-order position.** *See also:* **least significant character; least significant digit.** [610.5a]

LP. Acronym for **linear programming.** [610.2]

LSB. Acronym for **least significant bit.** [610.1]

Lukasiewicz notation. *See:* **prefix notation.** [610.1]

M. Abbreviation for **mega.** [610.1]

m. Abbreviation for **milli.** [610.1]

machine address.* *See:* **absolute address.**
*Deprecated. [610.12]

machine-aided. Pertaining to a process or function performed with the assistance of one or more computers. [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

machine-aided translation (MAT). Translation from one natural language to another, with the assistance of computer based aids such as automated lexicons and automated thesauri. *Syn:* **machine translation.** *See also:* **mechanical translation.** [610.2]

machine-centered simulation (machine-centred simulation). *See:* **computer simulation.** [610.3]

machine code. Computer instructions and data definitions expressed in a form that can be recognized by the processing unit of a computer. *Contrast with:* **assembly code; compiler code; interpretive code.** [610.12]

machine dependent. Pertaining to software that relies on features unique to a particular type of computer and therefore executes only on computers of that type. *Contrast with:* **machine independent.** [610.12]

machine independent. Pertaining to software that does not rely on features unique to a particular type of computer, and therefore executes on computers of more than one type. *Contrast with:* **machine dependent.** *See also:* **portability.** [610.12]

machine language. A language that can be recognized by the processing unit of a computer. Such a language usually consists of patterns of 1s and 0s, with no symbolic naming of operations or addresses. *Syn:* **first-generation language; machine-oriented language.** *Contrast with:* **assembly language; fifth-generation language; fourth generation language; high-order language; symbolic language.** [610.12]

machine-oriented language. *See:* **machine language.** [610.12]

machine readable. Pertaining to data in a form that can be automatically input to a computer; for example, data encoded on a diskette. [610.12]

MAchine-Readable Cataloging (MARC). In library automation, an internationally-accepted standard for systems used to create catalogs of machine-readable bibliographic records. *Syn:* **LCMARC; LCMARC.** [610.2]

machine-readable medium. A data medium that is machine-readable. *Syn:* **automated data medium.** [610.5]

machine recognition. *See:* **pattern recognition.** [610.4]

machine simulation. A simulation that is executed on a machine. *See also:* **computer simulation.** [610.3]

machine translation (MT). *See:* **machine-aided translation.** [610.2]

machine word. *See:* **word.** [610.1, 610.5]

macro. (1) In software engineering a predefined sequence of computer instructions that is inserted into a program, usually during assembly or compilation, at each place that its corresponding macroinstruction appears in the program. *Syn:* **macro definition.** *See also:* **macroinstruction; macrogenerator; open subroutine.** [610.12]
(2) In word processing, a predefined sequence of text and text formatting commands collected under a single user-defined name. Each time the name is entered, it is automatically replaced by the sequence of text and commands. [610.2]

macroassembler. An assembler that includes, or performs the functions of, a macrogenerator. [610.12]

macro definition. *See:* **macro.** [610.12]

macro generating program. *See:* **macrogenerator.** [610.12]

macrogenerator. A routine, often part of an assembler or compiler, that replaces each macroinstruction in a source program with the predefined sequence of instructions that the macroinstruction represents. *Syn:* **macro generating program.** [610.12]

macroinstruction. A source code instruction that is replaced by a predefined sequence of source instructions, usually in the same language as the rest of the program and usually during assembly or compilation. *See also:* **macro; macrogenerator.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

macro library. A collection of macros available for use by a macrogenerator. *See also:* **system library.** [610.12]

macroprocessor. A routine or set of routines provided in some assemblers and compilers to support the definition and use of macros. [610.12]

macroprogramming. Computer programming using macros and macroinstructions. [610.12]

magnetic character. *See:* **magnetic ink character.** [610.2]

magnetic ink. A special ink containing magnetic particles that can be detected and traced by input devices designed specifically for that purpose. [610.2]

magnetic ink character. A character imprinted on a document using magnetic ink. *Syn:* **magnetic character.** [610.2]

magnetic ink character recognition (MICR). The automatic recognition of magnetic ink characters. *See also:* **code for magnetic characters.** *Contrast with:* **optical character recognition.** [610.2]

Mahoney map. A diagram used in logic design, simplification, or optimization; invented by Matthew V. Mahoney. *See also:* **Karnaugh map; logic map.** [610.1]

mailbox service. *See:* **electronic mail.** [610.2]

main file. *See:* **master file.** [610.2, 610.5]

main program. A software component that is called by the operating system of a computer and that usually calls other software components. *See also:* **routine; subprogram.** [610.12]

maintainability. (1) The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment. *See also:* **extendability; flexibility.** [610.12]
(2) The ease with which a hardware system or component can be retained in, or restored to, a state in which it can perform its required functions. [610.12]

maintenance. (1) The process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment. *Syn:* **software maintenance.** *See also:* **adaptive maintenance; corrective maintenance; perfective maintenance.** [610.12]
(2) The process of retaining a hardware system or component in, or restoring it to, a state in which it can perform its required functions. *See also:* **preventive maintenance.** [610.12]

maintenance manual. *See:* **support manual.** [610.12]

majority. A Boolean operator having the property that if P is a statement, Q is a statement, R is a statement,..., then the majority of P,Q,R,... is true if more than half the statements are true, false if half or less are true. [610.1]

major key. *See:* **primary key.** [610.5]

management game. A simulation game in which participants seek to achieve a specified management objective given preestablished resources and constraints; for example, a simulation in which participants make decisions designed to maximize profit in a given business situation and a computer determines the results of those decisions. *See also:* **war game.** [610.3]

management information system (MIS). An automated system designed to provide managers with the information required to make basic decisions. *Syn:* **business information system; executive information system.** *See also:* **computer-aided management; decision support system.** [610.2]

man–centered simulation (man-centred simulation). *See:* **human-centered simulation.** [610.3]

man–machine interface (MMI). *See:* **user interface.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

man–machine simulation. *See:* **human-machine simulation.** [610.3]

mantissa. (1) The fractional part of a logarithm. *Contrast with:* **characteristic (1).** [610.1]

(2) For floating-point arithmetic, *see:* **significand.** [610.1]

manual hyphenation. In text formatting, hyphenation in which all line-ending and word break decisions are made by the user. *See also:* **automatic hyphenation; semi-manual hyphenation.** [610.2]

manufacture. In software engineering, the process of copying software to disks, chips, or other devices for distribution to customers or users. [610.12]

manufacturing phase. The period of time in the software life cycle during which the basic version of a software product is adapted to a specified set of operational environments and is distributed to a customer base. [610.12]

manuscript editor. *See:* **document editor.** [610.2]

many-to-many relationship. A relationship between two entities A and B such that any instance of A may be more associated with than one instance of B, and vice-versa. *Syn:* **m:n relationship.** [610.5]

map. (1) To establish a correspondence between the elements of one set and the elements of another. *Syn:* **map over.** [610.5]
(2) To establish a correspondence between the logical structure of a database and the physical structure of that database. [610.5]

map over. *See:* **map.** [610.5]

map program. A software tool, often part of a compiler or assembler, that generates a load map. [610.12]

MARC. Acronym for **MAchine Readable Cataloging.** [610.2]

margin-adjust hyphenation. *See:* **hot zone hyphenation.** [610.2]

margin-adjust zone. *See:* **hot zone.** [610.2]

mark. A symbol or group of symbols that indicates the beginning or end of a field, a word, an item of data, or a set of data such as a file, a record, or a block. [610.5]

mark detection. *See:* **mark sensing.** [610.2]

Markov chain. A discrete Markov process. [610.1a, 610.3]

Markov chain model. A discrete, stochastic model in which the probability that the model is in a given state at a certain time depends only on the value of the immediately preceding state. *Syn:* **Markov model.** *See also:* **semi-Markov model.** [610.1a, 610.3]

Markov model. *See:* **Markov chain model.** [610.3]

Markov process. A stochastic process which assumes that in a series of random events, the probability for occurrence of each event depends only on the immediately preceding outcome. *See also:* **semi-Markov process.** [610.3]

mark scanning. Optical sensing of marks recorded manually on a data medium. *Contrast with:* **mark sensing.** [610.2]

mark sensing. Electrical sensing of conductive marks recorded manually on a nonconductive data medium. For example, graphite marks on paper. *Syn:* **mark detection.** *Contrast with:* **mark scanning.** [610.2]

mark sensing column. A vertical line of positions on a data medium, capable of being detected by mark sensing. [610.2]

mask. A pattern of bits or characters designed to be logically combined with an unknown data item to retain or suppress portions of the data item; for example, the bit string "00000011" when logically ANDed with an eight-bit data item, gives a result that retains the last two bits of the data item and has zero in all the other bit positions. *See also:* **interrupt mask.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

mask document. In word processing, a form displayed on a display screen with blank areas for the user to complete. [610.2]

master file. An organized collection of records that is relatively permanent; for example, a file containing employee names, addresses, and salary information. *Syn:* **main file.** *Contrast with:* **transaction file.** [610.2a, 610.5]

master library. A software library containing master copies of software and documentation from which working copies can be made for distribution and use. *Contrast with:* **production library; software development library; software repository; system library.** [610.12]

master state. *See:* **supervisor state.** [610.12]

MAT. Abbreviation for **machine-aided translation.** [610.2]

match. (1) A condition in which the values of corresponding components of two or more data items are equal. *See also:* **hit.** [610.5] (2) To compare two or more data items to determine whether their corresponding components are equal as in (1). [610.5]

matched filtering. *See:* **template matching.** [610.4]

matching. *See:* **image matching.** [610.4]

matching interaction. An instruction method employed by some computer-assisted instruction systems, in which the student is asked to match answers to questions. *See also:* **question-and-answer interaction.** [610.2]

mathematical check. A check of the accuracy of a calculation by performing additional calculations. For example, verification of multiplication results by dividing the product by the multiplier to obtain the multiplicand. *Syn:* **arithmetic check.** [610.1]

mathematical model. A symbolic model whose properties are expressed in mathematical symbols and relationships; for example, a model of a nation's economy expressed as a set of equations. *Contrast with:* **graphical model; narrative model; software model; tabular model.** [610.3]

mathematical programming. In operations research, a procedure for locating the maximum or minimum of a function subject to constraints. [610.2]

matrix. A two-dimensional array, conceptually arranged in rows and columns. *Note:* A matrix with m rows and n columns is said to be of size $m \times n$ (m-by-n). *See also:* **column-major order; row-major order; table.** [610.5]

mean time between failures (MTBF). The expected or observed time between consecutive failures in a system or component. *See also:* **up time.** [610.12]

mean time to repair (MTTR). The expected or observed time required to repair a system or component and return it to normal operations. *See also:* **down time.** [610.12]

measurement space. In pattern recognition, a set of all possible n-tuples $(x_1, x_2, ..., x_n)$ that can be used to represent n measurements of a pattern. *See also:* **feature space.** [610.4]

measurement standard. A standard that describes the characteristics of evaluating a process of product. [610.12]

mechanical data processing. A method of data processing that involves the use of small, simple, mechanical machines. [610.2]

mechanical inspection. *See:* **computer-aided inspection.** [610.2]

mechanical translation. Translation from one natural language to another by a computer or through some other mechanical means. *See also:* **machine-aided translation.** [610.2]

medical information system (MIS). *See:* **hospital information system.** [610.2]

medium. *See:* **data medium; empty medium; machine-readable medium; virgin medium.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

meet. *See:* **AND.** [610.1]

mega (M). (1) A prefix indicating one million (10^6). [610.1]
(2) In statements involving size of computer storage, a prefix indicating 2^{20}, or 1,048,576. [610.1]

member. In data management, a subunit contained in a partitioned data set. [610.5]

memory capacity. The maximum number of items that can be held in a given computer memory; usually measured in words or bytes. *See also:* **channel capacity; storage capacity.** [610.12]

memory compaction. (1) A storage allocation technique in which the contents of all allocated storage areas are moved to the beginning of the storage space and the remaining storage blocks are combined into a single block. *Syn:* **garbage collection.** [610.12]
(2) A storage allocation technique in which contiguous blocks of nonallocated storage are combined to form single blocks. [610.12]

memory dump. A display of the contents of all or part of a computer's internal storage, usually in binary, octal, or hexadecimal form. *See also:* **change dump; dynamic dump; port mortem dump; selective dump; snapshot dump; static dump.** [610.12]

memory map. A diagram that shows where programs and data are stored in a computer's memory. [610.12]

menu. A list of options available for selection by the user of a computer system. *See also:* **help menu; menu selection.** [610.2]

menu by-pass. In a menu-driven system, a feature that permits advanced users to perform functions in a command-driven mode without selecting options from the menus. [610.12]

menu-driven. Pertaining to a system or mode of operation in which the user directs the system through menu selections. *See also:* **menu by-pass.** *Contrast with:* **command-driven.** [610.12]

menu selection. (1) The process of choosing an item from a menu. [610.2]
(2) The item chosen from a menu. [610.2]

merge. To combine the items of two or more sets, all in the same order, into one set in that order. *See also:* **balanced merge; bitonic merge; coalesce; collate; merge sort; order-by-merging; unbalanced merge.** [610.5]

merge exchange sort. *See:* **Batcher's parallel sort.** [610.5]

merge search. A sequential search in which the set of search arguments is ordered in the same sequence as the set to be searched; the set is searched sequentially, using the first search argument, until an equal or greater search key is found, the former case signifying a successful search, the latter, an unsuccessful search; the search for the next search argument begins where the last search left off. [610.5]

merge sort. A sort in which the set to be sorted is divided into subsets, the items in each subset are sorted, and the sorted subsets are merged. *Syn:* **merging sort.** *See also:* **external merge sort; internal merge sort.** [610.5]

merging sort. *See:* **merge sort.** [610.5]

metacompiler. *See:* **compiler generator.** [610.12]

metadata. Data that describes other data; for example, a data dictionary contains a collection of metadata. [610.5]

metalanguage. A language used to specify some or all aspects of a language; for example, Backus-Naur form. *See also:* **stratified language; unstratified language.** [610.12]

method standard. A standard that describes the characteristics of the orderly process or procedure used in the engineering of a product or performing a service. [610.12]

metric. A quantitative measure of the degree to which a system, component, or process possesses a given attribute. *See also:* **quality metric.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

MFLOPS. Acronym for millions of floating point operations per second. A measure of computer processing speed. *See also:* **KOPS; MIPS.** [610.12]

MICR. Acronym for **magnetic ink character recognition.** [610.2]

micro. A prefix indicating one millionth (10^{-6}). [610.1]

microarchitecture. The microword definition, data flow, timing constraints, and precedence constraints that characterize a given microprogrammed computer. [610.12]

microcode. A collection of microinstructions, comprising part of, all of, or a set of microprograms. [610.12]

microcode assembler. A computer program that translates microprograms from symbolic form to binary form. [610.12]

microcopy. A copy of an image or document so reduced in size from its original that it cannot be read by the unaided human eye. For example, microform, microfiche, microfilm, microimage. [610.2]

microfacsimile. Transmission and reception of microimages via facsimile communication. [610.2]

microfiche. A sheet of microfilm capable of containing microimages in a grid pattern. The sheet usually contains a title that can be read without magnification. *Syn:* **fiche.** [610.2]

microfilm. (1) A high resolution film for recording microimages. [610.2] (2) To record microimages on film. *See also:* **computer output microfilm.** [610.2]

microfilmer. *See:* **computer output microfilmer.** [610.2]

microfont.* *See:* **optical character recognition-B.** [610.2]
* Deprecated.

microform. A medium that contains microimages. For example, microfiche, microfilm. [610.2]

micrographics. That branch of science and technology concerned with methods and techniques for converting information to or from microform. *Syn:* **microphotographics.** *See also:* **office automation.** [610.2]

microimage. An image that is too small to be read by the human eye without magnification. [610.2]

microinstruction. In microprogramming, an instruction that specifies one or more of the basic operations needed to carry out a machine language instruction. Types include diagonal microinstruction; horizontal microinstruction; vertical microinstruction. *See also:* **microcode; microoperation; microprogram.** [610.12]

microoperation. In microprogramming, one of the basic operations needed to carry out a machine language instruction. *See also:* **microinstruction.** [610.12]

microphotographics. *See:* **micrographics.** [610.2]

microprint. A positive microcopy photographically printed onto paper. [610.2]

microprogram. A sequence of instructions, called microinstructions, specifying the basic operations needed to carry out a machine language instruction. [610.12]

microprogrammable computer. A microprogrammed computer in which microprograms can be created or altered by the user. [610.12]

microprogrammed computer. A computer in which machine language instructions are implemented by microprograms rather than by hard-wired logic. *Note:* A microprogrammed computer may or may not be a microcomputer; the concepts are not related despite the similarity of the terms. *See also:* **microarchitecture; microprogrammable computer.** [610.12]

microprogramming. The process of designing and implementing the control logic of a computer by identifying the basic operations needed to carry out each machine language

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052872 9

instruction and representing these operations as sequences of instructions in a special memory called control store. This method is an alternative to hard wiring the control signals necessary to carry out each machine language instruction. Techniques include bit steering, compaction, residual control, single-level encoding, two-level encoding. *See also:* **microcode; microinstruction; microprogram.** [610.12]

micropublishing. The production and distribution of information via microform. The information may be original or may have been previously published in another form. [610.2]

microword. An addressable element in the control store of a microprogrammed computer. [610.12]

mid-square function. In hashing, a hash function that returns the middle digits of the square of the original key. For example, in the function below, the middle three digits are returned. *Syn:* **middle-square function.** [610.5]

Original key	Calculation	Hash Value
2964	2964 x 2964 = 8,785,296	852
1119	1119 x 1119 = 110,781	78

middle-square function. *See:* **mid-square function.** [610.5]

milli (m). A prefix indicating one thousandth (10^{-3}). [610.1]

minimum delay programming. A programming technique in which storage locations for computer instructions and data are chosen so that access time is minimized. [610.12]

minimum-distance code. A BCD code in which the Hamming distance between consecutive numerals does not fall below a specified minimum value. [610.1]

minor key. *See:* **secondary key.** [610.5]

minuend. A number from which another number (the subtrahend) is subtracted to produce a result (the difference). [610.1]

minus. *See:* **difference.** [610.5]

MIPS. Acronym for million instructions per second. A measure of computer processing speed. *See also:* **KOPS; MFLOPS.** [610.12]

MIS. (1) Acronym for **management information system.** [610.2]
(2) Acronym for medical information system. *See:* **hospital information system.** [610.2]

misdetection. In pattern recognition, the failure to detect the existence of a pattern. [610.4]

misidentification. In pattern classification, the failure to assign a pattern to its true pattern class. *Syn:* **type I error.** *Contrast with:* **false identification.** [610.4]

mistake. A human action that produces an incorrect result. *Note:* The fault tolerance discipline distinguishes between the human action (a mistake), its manifestation (a hardware or software fault), the result of the fault (a failure), and the amount by which the result is incorrect (the error). [610.12]

mixed mode. Pertaining to an expression that contains two or more different data types. For example, $Y := X + N$, where X and Y are floating point variables and N is an integer variable. *Syn:* **mixed type.** [610.12]

mixed type. *See:* **mixed mode.** [610.12]

mixed-base notation. *See:* **mixed-radix notation.** [610.1]

mixed-base numeration system. *See:* **mixed-radix notation.** [610.1]

mixed-radix notation. A radix notation system in which all digit positions do not have the same radix. For example, biquinary notation in which the digit positions have the radix 2 or 5, alternately. *Syn:* **mixed-base notation; mixed-base numeration system; mixed-radix numeration system.** *Contrast with:* **fixed-radix notation.** [610.1]

mixed-radix numeration system. *See:* **mixed-radix notation.** [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

MMI. Acronym for **man-machine interface.** *See:* **user interface.** [610.12]

m:n relationship. * *See*: **many-to-many relationship.** [610.5]
* Deprecated.

mock-up. A full-sized structural, but not necessarily functional, model built accurately to scale, used chiefly for study, testing, or display; for example, a full-sized model of an airplane displayed in a museum. *See also:* **physical model.** [610.3]

mod. Acronym for **modulo.** [610.1]

mode. A variable that a user may set, sense, save, and restore to control the execution of subsequent arithmetic operations. [610.1]

model. (1) An approximation, representation, or idealization of selected aspects of the structure, behavior, operation, or other characteristics of a real-world process, concept, or system. *Note:* Models may have other models as components. [610.3]
(2) To serve as a model as in (1). [610.3]
(3) To develop or use a model as in (1). [610.3]

model validation. The process of determining the degree to which the requirements, design, or implementation of a model are a realization of selected aspects of the system being modeled. *See also:* **fidelity.** *Contrast with:* **model verification.** [610.3]

model verification. The process of determining the degree of similarity between the realization steps of a model; for example, between the requirements and the design, or between the design and its implementation. *Contrast with:* **model validation.** [610.3]

modify. (1) To change the contents of a database. [610.5]
(2) To change the logical structure of a database. *See also*: **alter.** [610.5]

modular. Composed of discrete parts. *See also:* **modular decomposition; modular programming.** [610.12]

modular decomposition. The process of breaking a system into components to facilitate design and development; an element of modular programming. *Syn:* **modularization.** *See also:* **cohesion; coupling; demodularization; factoring; functional decomposition; hierarchical decomposition; packaging.** [610.12]

modularity. The degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components. *See also:* **cohesion; coupling.** [610.12]

modularization. *See:* **modular decomposition.** [610.12]

modular programming. A software development technique in which software is developed as a collection of modules. *See also:* **data structure-centered design; input-process-output; modular decomposition; object-oriented design; rapid prototyping; stepwise refinement; structured design; transaction analysis; transform analysis.** [610.12]

module. (1) A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to, or output from, an assembler, compiler, linkage editor, or executive routine. [610.12]
(2) A logically separable part of a program. *Note:* The terms "module," "component," and "unit" are often used interchangeably or defined to be sub-elements of one another in different ways depending upon the context. The relationship of these terms is not yet standardized. [610.12]

module counter. A counter that reverts to zero in the counting sequence after reaching a value of n-1. [610.1]

module strength. *See:* **cohesion.** [610.12]

module testing. *See:* **component testing.** [610.12]

modulo (mod). An arithmetic operation that yields the remainder of an integer division problem. For example 39-3 modulo 6. [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052874 2

modulo-*n* residue. The remainder obtained by dividing a number by *n*. [610.1]

modulo-two sum. *See:* **exclusive OR.** [610.1]

modulus. The number of integers that can be represented in a numeration system. For example, in a system with a modulus of five, the only integers that can be represented are 0, 1, 2, 3, and 4. [610.1]

molecular data element. *See:* **composite data element.** [610.5]

monadic. Pertaining to an operation involving a single operand. *Contrast with:* **dyadic.** [610.1]

monadic Boolean operation. A logical operation involving one operand. For example, the NOT operation. *Contrast with:* **dyadic Boolean operation.** [610.1]

monadic operation. An operation involving one operand. For example, the square root operation. *Syn:* **unary operation.** *Contrast with:* **dyadic operation.** [610.1]

monadic operator. An operator that specifies an operation on one operand. For example, the square root operator. *Syn:* **unary operator.** *Contrast with:* **dyadic operator.** [610.1]

monadic selective construct. An if-then-else construct in which processing is specified for only one outcome of the branch, the other outcome resulting in skipping this processing. *Contrast with:* **dyadic selective construct.** [610.12]

monitor. A software tool or hardware device that operates concurrently with a system or component and supervises, records, analyzes, or verifies the operation of the system or component. *Syn:* **execution monitor.** *See also:* **hardware monitor; software monitor.** [610.12]

Monte Carlo method. In modeling and simulation, any method that employs Monte Carlo simulation to determine estimates for unknown values in a deterministic problem. [610.1a, 610.3]

Monte Carlo simulation. A deterministic simulation in which random statistical sampling techniques are employed such that the result determines estimates for unknown values. [610.3]

month-end processing. The operations required to complete a monthly cycle. For example, monthly ledger processing. [610.2]

monthly cycle. One complete execution of a data processing function that must be performed once a month. *See also:* **annual cycle; daily cycle; quarterly cycle; weekly cycle.** [610.2]

mortising. *See:* **kerning.** [610.2]

most significant bit (MSB). The bit having the greatest effect on the value of a binary numeral; usually the leftmost bit. [610.1]

most significant character. The character in the leftmost position in a character string. *Contrast with:* **least significant character.** [610.5]

most significant digit. The digit having the greatest effect on the value of a numeral; usually the leftmost digit; for example, the 7 in 756.4. *Contrast with:* **least significant digit.** [610.1a, 610.5]

motion picture display. *See:* **cine-oriented image.** [610.2]

***m*-out-of-*n* code.** A binary code in which *m* of the *n* digits that represent a word, character, or digit are in one state, and the other digits are in the opposite state. *See also:* **two-out-of five code.** [610.1]

move. (1) To read data from a source, altering the contents of the source location, and to write the same data elsewhere in a physical form that may differ from that of the source. For example, to move data from one file to another. *Contrast with:* **copy.** [610.12]
(2) Sometimes, a synonym for **copy.** [610.12]
See also: **fetch; load; store.** [610.12]

moving average. An average calculated on a selected, changing subset of a time series of data. For example, a four-point moving

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

average would be the average of the last four data points in the time series. [610.1]

MSB. Acronym for **most significant bit.** [610.1]

MT. Acronym for **machine translation.** *See:* **machine-aided translation.** [610.2]

MTBF. Acronym for **mean time between failures.** [610.12]

MTTR. Acronym for **mean time to repair.** [610.12]

multiaddress instruction. A computer instruction that contains more than one address field. *Syn:* **multiple-address instruction.** *Contrast with:* **one-address instruction.** [610.12]

multi-band image. A set of images of the same scene, each formed by radiation from a different segment of the spectrum. [610.4]

multifield key. *See*: **concatenated key.** [610.5]

multilevel address. *See:* **indirect address.** [610.12]

multilevel storage. *See:* **virtual storage.** [610.12]

multilist. A technique for organizing records in which records that have equivalent values for a given secondary key form a linked list. *Contrast with*: **inverted list (2).** *Syn*: **multiple threaded list.** [610.5]

Student	Name	Homeroom	Link
1	MARY	25	4
2	JOE	27	15
3	JOHN	10	6
4	ANNE	25	5
5	SUSAN	25	-
6	KIM	10	21
7	BOB	26	16
.	.	.	.

Fig 49
Multilist

multiple-address instruction. *See:* **multiaddress instructions.** [610.12]

multiple arithmetic. A system or method of performing ordinary arithmetic with a digital computer where several parts of one or more numbers are utilized in an arithmetic operation, yielding several results. [610.1]

multiple choice interaction. An instruction method employed by some computer-assisted instruction systems, in which the student is asked to choose one of a set of multiple choice answers in response to a question. *See also:* **question-and-answer interaction.** [610.2]

multiple exclusive selective construct. *See:* **case.** [610.12]

multiple inclusive selective construct. A special instance of the case construct in which two or more different values of the control expression result in the same processing. For example, values 1 and 2 cause one branch, 3 and 4 cause another, and so on. [610.12]

multiple list insertion sort. *See*: **address calculation sort.** [610.5]

multiple precision. Pertaining to the use of two or more computer words to represent a number in order to preserve or gain precision. *Contrast with*: **single precision.** *Syn:* **extended precision; multiprecision.** *See also*: **double precision; triple precision.** [610.1, 610.5]

multiple-precision arithmetic. Computer arithmetic performed with operands that are expressed in multiple-precision representation. [610.1]

multiple threaded list. *See*: **multilist.** [610.5]

multiplicand. A number to be multiplied by another number (the multiplier) to produce a result (the product). [610.1]

multiplication transformation function. In hashing, a hash function that returns the original key multiplied by some value. For example, in the function below, the original

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052876 6

key is multiplied by the length of the record in which it is found. *See also*: **mid-square function.** [610.5]

Original Record	Calculation	Hash Value
35 Bob White	35 x 13 = 448	448
41 Richard Doe	41 x 17 = 697	697

multiplier. A number by which another number (the multiplicand) is multiplied to produce a result (the product). [610.1]

multiprecision. *See:* **multiple precision.** [610.1]

multiprocessing. A mode of operation in which two or more processes are executed concurrently by separate processing units that have access (usually) to a common main storage. *Contrast with:* **multiprogramming.** *See also:* **multitasking; time sharing.** [610.12]

multiprogramming. A mode of operation in which two or more computer programs are executed in an interleaved manner by a single processing unit. *Contrast with:* **multiprocessing.** *See also:* **multitasking; time sharing.** [610.12]

multitasking. A mode of operation in which two or more tasks are executed in an interleaved manner. *See also:* **multiprocessing; multiprogramming; time sharing.** [610.12]

multivalued dependency. A type of dependency among three attributes A, B, and C in relation R, in which B is multivalued dependent on A if, and only if, the set of values of B that match a given pair of values for A and C depends only on the value for A and is independent of the value for C. *See also*: **fourth normal form.** [610.5]

multiway merge sort. A merge sort in which the set to be sorted is divided into two or more ordered subsets that are merged by comparing the smallest items of each subset, outputting the smallest of those, then repeating the process. *See also*: **two-way merge sort.** [610.5]

multiway radix trie search. A radix trie search using a trie of order greater than 2, in which more than one digit is considered on each branch. *See also*: **binary radix trie search.** [610.5]

multiway tree. A tree of order greater than 2. [610.5]

mutation. *See:* **program mutation.** [610.12]

mutation testing. A testing methodology in which two or more program mutations are executed using the same test cases to evaluate the ability of the test cases to detect differences in the mutations. [610.12]

n. Abbreviation for **nano.** [610.1a]

***n*-address instruction.** A computer instruction that contains n address fields, where *n* may be any non-negative integer. *See also:* **one-address instruction; two-address instruction; etc.** *Contrast with:* ***n*-plus-one address instruction.** [610.12]

***n*-adic.** *See:* ***n*-ary.** [610.1]

***n*-adic Boolean operation.** A Boolean operation involving exactly *n* operands. *Syn:* ***n*-ary Boolean operation.** *See also:* **dyadic Boolean operation; monadic Boolean operation.** [610.1]

***n*-adic operation.** An operation involving exactly *n* operands. *Syn:* ***n*-ary operation.** *See also:* **dyadic operation; monadic operation.** [610.1]

nano (n). A prefix indicating 10^{-9}. [610.1a]

***n*-ary.** (1) Pertaining to a selection in which there are *n* possible outcomes. [610.1] (2) Pertaining to a numeration system with a radix of *n*. *Syn:* ***n*-adic.** [610.1]

n*-ary Boolean operation. *See:* ***n*-adic Boolean operation.**
* Deprecated. [610.1]

***n*-ary operation.** *See:* ***n*-adic operation.** [610.1]

***n*-ary relation.** A relation with *n* attributes in each tuple. [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

***n*-ary tree.** A tree of order n; for example, an 8-ary tree. *See also*: **binary tree; multiway tree; octary tree; quadary tree.** [610.5]

***n*-bit byte.** A group of *n* adjacent binary digits operated upon as a unit. [610.1]

NAK. Acronym for the **negative acknowledge character.** [610.5]

name. An alphanumeric term that identifies a data item such as a field, record, or file. [610.5]

NaN. Acronym for not a number; a symbolic entity encoded in floating-point format. [610.1]

NAND. A Boolean operator having the property that if P is a statement, Q is a statement, R is a statement,..., then the NAND of P, Q, R,... is true if and only if at least one statement is false. *Note:* The NAND of P and Q is often represented by $\overline{PQ}$. *Syn:* **nonconjunction; Sheffer stroke.** [610.1]

P	Q	$\overline{PQ}$
0	0	1
0	1	1
1	0	1
1	1	0

Fig 50
NAND Truth Table

nanocode. A collection of nanoinstructions. [610.12]

nanoinstruction. In a two-level implementation of microprogramming, an instruction that specifies one or more of the basic operations needed to carry out a microinstruction. [610.12]

nanostore. In a two-level implementation of microprogramming, a secondary control store in which nanoinstructions reside. [610.12]

narrative information. Information that is presented according to the syntactic order of a natural language. *Contrast with*: **formatted information.** [610.5]

narrative model. A symbolic model whose properties are expressed in words; for example, a written specification for a computer system. *Syn:* **verbal-descriptive model.** *Contrast with:* **graphical model; mathematical model; software model; tabular model.** [610.3]

Nassi-Shneiderman chart. *See:* **box diagram.** [610.12]

national character. Deprecated term for the characters #, @, and $. [610.5]

native data type. A data type that is built into a software or hardware system. [610.5]

natural binary. *See:* **binary.** [610.1]

natural language. A language whose rules are based on usage rather than being preestablished prior to the language's use. Examples include German and English. *Contrast with:* **formal language.** [610.12]

natural model. A model that represents a system by another system that already exists in the real world; for example, a model that uses one body of water to represent another. [610.3]

natural number. (1) A non-negative integer: {0, 1, 2,...} [610.1a]
(2) A positive integer {1, 2, 3,...}. [610.1, 610.5a]

natural two-way merge sort. A two-way merge sort in which the set to be sorted is repeatedly divided into two ordered subsets and merged, taking advantage of runs which occur naturally in the input set. *Contrast with*: **straight two-way merge sort.** [610.5]

NC (N/C). Acronym for **numerical control.** [610.2]

negate. To perform the NOT operation. [610.1]

negation. *See:* **NOT.** [610.1]

negative acknowledge character (NAK). A transmission control character transmitted by a station as a negative response to the station with which the connection has been set up. [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

neighborhood. In image processing, a set of pixels located near a given pixel. [610.4]

neighborhood operator. An image operator that assigns a gray level to each output pixel based on the gray levels in a neighborhood of the corresponding input pixel. *Contrast with:* **point operator.** [610.4]

NEITHER-NOR.* *See:* **NOR.** [610.1]
*Deprecated.

nest. To incorporate a computer program construct into another construct of the same kind. For example, to nest one subroutine, block, or loop within another; to nest one data structure within another. [610.12]

net structure. *See:* **network structure.** [610.5]

network. (1) A data structure in which components are allowed to have more than one superordinate component. *See also:* **hierarchy; graph.** [610.5]
(2) A graph in which the edges connecting the nodes are assigned weights representing some characteristic, such as cost or quantity, related to the edge. [610.5]

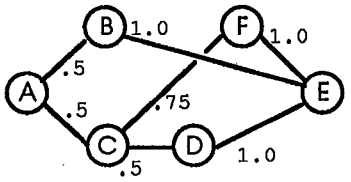


Fig 51
Network

network database. A database in which data are organized into segments which represent nodes within a network. *Contrast with:* **hierarchical database; relational database.** [610.5]

network model. A data model in which entities are represented as nodes within a modified tree structure that permits all but the root to have multiple parents. *Note:* The network model was originally proposed as the CODASYL model by the Conference on Data Systems Languages (CODASYL) and described in their 1968 and 1971 publications. [610.5]

network structure. A collection of entities that are organized in a network fashion. *Contrast with:* **hierarchical structure.** *Syn:* **net structure.** [610.5]

network theory. The study of networks used to model processes such as communications, computer performance, routing problems, and project management. [610.3]

new-line character (NL). *See:* **carriage return character.** [610.5]

nibble (nybble). *See:* **quartet.** [610.1]

nines check. *See:* **casting out nines.** [610.1]

nines complement. The diminished-radix complement of a decimal numeral, which is formed by subtracting each digit from 9. For example, the nines complement of 4830 is 5169. *Syn:* **complement on nine.** [610.1]

NL. Acronym for the **new-line character.** [610.5]

***n*-level address.** An indirect address that specifies the first of a chain of n storage locations, the first n-1 of which contain the address of the next location in the chain and the last of which contains the desired operand. For example, a two-level address. *Contrast with:* **direct address; immediate data.** [610.12]

***n*-*m* tree.** A tree in which each node has at least n but no more than m subtrees; for example, 2-4 tree. *Note:* A 2-4 tree is sometimes written as 2-3-4 tree. [610.5]

no-fill mode. In text formatting, an operating mode in which no justification is performed. [610.2]

no-op. Abbreviation for **no-operation.** [610.12]

no-operation. A computer operation whose execution has no effect except to advance the instruction counter to the next instruction. Used to reserve space in a program or, if executed repeatedly, to wait for a given event. Often abbreviated no-op. *Syn:* **do-nothing operation.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

nodalization. (1) The set of nodes within a system being modeled. [610.3]
(2) The process of developing the nodes as in (1). [610.3]

node. (1) In a diagram, a point, circle, or other geometric figure used to represent a state, event, or other item of interest. *See also:* graph (2). [610.12]
(2) A single entity that is represented in a mathematical model; for example, in a model of a nuclear reactor, a water pump or section of pipe. [610.3]
(3) In a tree, an element that is used to contain information that describes some object; for which there is at least one key used to identify the node. *Note*: Nodes are connected to each other by link fields to form the tree. *Syn*: **vertex.** *See also*: **child node; nonterminal node; parent node; root node; terminal node.** [610.5]
(4) *Note:* The meaning of this term in the context of computer networks will be covered in P610.7. [610.12]

noise. Irrelevant data that hamper recognition and interpretation of data of interest. [610.4]

noise cleaning. *See:* **smoothing.** [610.4]

noise suppression. *See:* **smoothing.** [610.4]

nomenclature standard. A standard that describes the characteristics of a system or set of names, or designations, or symbols. [610.12]

nonabsorbing state. In a Markov chain model, a state that can be left once it is entered. *Contrast with:* **absorbing state.** [610.3]

nonarithmetic shift. *See:* **logical shift.** [610.1]

nonconjunction. *See:* **HAND.** [610.1]

nondestructive addition. Addition performed on a computer in such a manner that the first operand placed in the arithmetic register is the augend. The addend is then added, and the sum becomes the new augend. *Contrast with:* **destructive addition.** [610.1]

nondestructive read. A read operation that does not erase the data in the accessed location. *Contrast with:* **destructive read.** [610.12]

nondisjunction. *See:* **NOR.** [610.1]

nonequivalence. *See:* **exclusive OR.** [610.1]

nonidentity operation. A Boolean operation whose result is true if and only if not all of the operands have the same Boolean value. *Note:* A nonidentity operation on two operands is the same as an exclusive-OR operation. [610.1]

nonlinear data structure. A nonprimitive data structure that can represent data that is multidimensional in nature. For example, a tree. *Contrast with*: **linear data structure.** [610.5]

nonlinear optimization. *See:* **nonlinear programming.** [610.2]

nonlinear programming. In operations research, a procedure for locating the maximum or minimum of a function of variables that are subject to constraints, when either the function or the constraints, or both, are nonlinear. *Syn:* **nonlinear optimization.** *Contrast with:* **linear programming.** *See also:* **convex programming; quadratic programming.** [610.2]

nonlocking. In code extension characters, having the characteristic that a change in interpretation applies only to a specified number of the coded representations following, commonly only one. *Contrast with*: **locking.** [610.5]

nonlocking shift character. A shift-out character that causes the character following it to be interpreted as a member of a different character set from the original set. *Contrast with*: **locking shift character.** [610.5]

nonprime attribute. An attribute that is not part of any candidate key of a relation. *Contrast with*: **prime attribute.** [610.5]

nonprimitive data structure. A structured set of primitive data structures. Structures may be linear, as in a vector, or nonlinear as in a tree. *See also*: **linear data structure; nonlinear data structure.** *Syn*: **complex data structure.** *Contrast with*: **primitive data structure.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052880 8

nonprocedural language. A language in which the user states what is to be achieved without having to state specific instructions that the computer must execute in a given sequence. *Contrast with:* **procedural language.** *See also:* **declarative language; interactive language; rule-based language.** [610.12]

nonsignificant code. A code that identifies a particular item but does not yield further information about the properties or classification of the item. *Contrast with*: **significant code.** [610.5]

nonterminal node. In a tree, a node that can have one or more subtrees. *Syn*: **branch node; internal node.** *Contrast with*: **terminal node.** *See also*: **root node.** [610.5]

nontransitive dependency. A type of dependency among attributes in a relation, in which a nonprime attribute A is said to be nontransitively dependent on another attribute B if and only if A is dependent on B, and there is another attribute C that is functionally dependent on B but does not functionally determine A. *Contrast with*: **transitive dependency.** [610.5]

NOR. (1) A Boolean operator having the property that if P is a statement, Q is a statement, R is a statement,..., then the NOR of P,Q,R,... is true if and only if all statements are false. *Note:* P NOR Q is often represented by P↓Q. *Syn:* **nondisjunction.** [610.1]
(2) In configuration management, an acronym for **notice of revision.** [610.12]

P	Q	P↓Q
0	0	1
0	1	0
1	0	0
1	1	0

Fig 52
NOR (1) Truth Table

normal binary. *See:* **binary.** [610.1]

normal form. The form of a data structure, relation, or database that has been reduced to a simpler, more stable form than it was in its unnormalized form. *See also*: **Boyce/Codd normal form; first normal form; fourth normal form; projection/join normal form; second normal form; third normal form.** [610.5]

normal number. A non-zero number that is finite and not subnormal. [610.1]

normal random number. Any member of a random number sequence that has a normal, or Gaussian, distribution. [610.1]

normalization. (1) The process of decomposing and restructuring a complex data structure in order to reduce the structure to a simpler, more stable form. *Note*: Such a data structure is said to be in "normal form." [610.5]
(2) The process of reducing a relation to its simplest form such that each attribute is derived from a single domain consisting of nondecomposable values. [610.5]

normalize. (1) In database design, to reduce a data structure, relation, or database to a simpler, more stable form. *Syn*: **standardize.** *See also*: **normal form; normalized form.** [610.5]
(2) To alter or position data into a standard format, as in justification of text. [610.5]
(3) To adjust the exponent and mantissa of floating-point data such that the mantissa lies in a standard range. *See:* **normalized form.** [610.5]
(4) To shift the fixed-point part of a floating-point number, and make the corresponding adjustment to the exponent, to ensure that the fixed-point part lies within some prescribed range. The number represented remains unchanged. *Syn:* **standardize.** [610.1]

normalized form. (1) In database design, the form assumed by data that have been normalized. *See also:* **normal form.** *Contrast with:* **unnormalized form.** [610.5]
(2) The form taken by a floating-point representation when the fixed-point part lies within some standard range, so chosen that any given real number can be represented by a unique pair of numerals. Examples

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

below illustrate real data and their corresponding normalized form such that the fixed-point portion is in the form x.xxx. [610.5]

$.123 * 10^4 = 1.23 * 10^3$
$.999 * 10^{-1} = 9.99 * 10^{-2}$

Syn: **standard data.** [610.5]

normalized relation. *Contrast with*: **unnormalized relation.** *See*: **normalize; relation.** [610.5]

normative model. A model that makes use of a familiar situation to represent a less familiar one; for example, a model that depicts the human cardiovascular system by using a mechanical pump, rubber hoses, and water. [610.3]

NOT. A monadic Boolean operator having the property that if P is a statement, then the expression "NOT P" is true if P is false, and false if P is true. *Note:* NOT P is often represented by ~P, $\bar{P}$, or P′. *Syn:* **Boolean complementation; complementary operator; inversion (1); negation.** [610.1]

P	$\bar{P}$
0	1
1	0

Fig 53
NOT Truth Table

NOT-AND.* *See:* **NAND.** [610.1]
*Deprecated.

notation. A system of symbols used to represent information, and the rules for their use. [610.1]

notation standard. A standard that describes the characteristics of formal interfaces within a profession. [610.12]

NOT-BOTH. *See:* **NAND.** [610.1]
*Deprecated.

notice of revision (NOR). A form used in configuration management to propose revisions to a drawing or list, and, after approval, to notify users that the drawing or list has been, or will be, revised accordingly. *See also:* **configuration control; engineering change; specification change notice.** [610.12]

NOT-IF-THEN. *See:* **exclusion.** [610.1]

NOT-OR.* *See:* **NOR.** [610.1]
*Deprecated.

noughts complement. *See:* **radix complement.** [610.1]

novenary. (1) Pertaining to a selection in which there are nine possible out-comes. [610.1]
(2) Pertaining to the numeration system with a radix of 9. [610.1]

novendenary. (1) Pertaining to a selection in which there are 19 possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of 19. [610.1]

***n*-plus-one address instruction.** A computer instruction that contains $n+1$ address fields, the last containing the address of the instruction to be executed next. *See also:* **one-plus-one address instruction; two-plus-one address instruction; etc.** *Contrast with:* ***n*-address instruction.** [610.12]

***n*-tuple.** (1) An ordered set of values $(x_1, x_2, ..., x_n)$. [610.1, 610.5]
(2) In a relational data model, a tuple. [610.5]

nucleus. *See:* **kernel (1).** [610.12]

NUL. Abbreviation for the **null character.** [610.5]

null character (NUL). A control character that is used to accomplish media-fill or time-fill, and that may be inserted into or removed from, a sequence of characters without affecting the meaning of the sequence; however, the control of equipment or the format may be affected by this character. *See also:* **space character.** [610.5]

null data. Data for which space is allocated but for which no value currently exists. [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

null pointer. A pointer that is empty; that is, a pointer that does not point to anything. [610.5]

null string. A string containing no entries. *Note*: It is said that a null string has length zero. [610.5]

null transaction. A transaction that has no effect on the master file that is being updated. It is usually used for documentation purposes only. *See also:* **add transaction; change transaction; delete transaction; update transaction.** [610.2]

null tree. A tree that has exactly one root and one descendant node. [610.5]

number. (1) A mathematical abstraction indicating a quantity or amount. [610.1]
(2) Loosely, a numeral. [610.1]

number crunching. Computer processing that relies heavily on the arithmetic and logical capabilities of the central processing unit, as contrasted with processing that entails extensive input/output or data movement. [610.2]

number range. The set of values that a number may assume. [610.1]

number representation. A representation of a number in a numeration system. *Syn:* **numeration.** [610.1]

number representation system. A system for the representation of numbers; for example, the decimal numeration system, the Roman numeral system, the binary numeration system. *Syn:* **numeration system; numeral system.** [610.1]

number system. Loosely, a numeration system. [610.1]

numeral. A representation of a number. *See also:* **binary numeral; decimal numeral; octal numeral; hexadecimal numeral.** [610.1]

numeral system. *See:* **number representation system.** [610.1]

numeration. *See:* **number representation.** [610.1]

numeration system. *See:* **number representation system.** [610.1]

numeric. Pertaining to data that can be expressed using only numbers and mathematical symbols, in contrast to characters or other special signs or symbols. *Syn*: **numerical.** *See also*: **arithmetic; numeric data; pure numeric.** [610.1a, 610.5]

numerical. *See*: **numeric.** [610.1, 610.5]

numerical analysis. The study of methods of obtaining useful quantitative solutions to problems that have been expressed mathematically, including the study of the errors and bounds on errors in obtaining such solutions. [610.1]

numerical control (NC or N/C). Computer control of machines that produce manufactured parts. *See also:* **computer numerical control; direct numerical control; distributed numerical control; process control.** [610.2]

numerical control machine. A machine that produces manufactured parts under automatic control. *Syn:* **numerical control tool.** [610.2]

numerical control tool. *See:* **numerical control machine.** [610.2]

numerical model. (1) A mathematical model in which a set of mathematical operations are reduced to a form suitable for solution by a simpler methods such as numerical analysis or automation; for example, a model in which a single equation representing a nation's economy is replaced by a large set of simple averages based on empirical observations of inflation rate, unemployment rate, gross national product, and other indicators. [610.3]
(2) A model whose properties are expressed by numbers. [610.3]

numerical shift. *See:* **arithmetic shift.** [610.1]

numeric bit data. *See*: **binary picture data.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

numeric character. *See*: **digit.** [610.1, 610.5]

numeric character data. *See*: **decimal picture data.** [610.5]

numeric character set. A character set that contains digits and may contain control characters, special characters, and the space character, but not letters. [610.5]

numeric code. A code that uses numerals to represent data. [610.1, 610.5]

numeric data. Data used to represent numbers. *See also*: **binary data; complex data; integer data; packed data; real data.** [610.5]

numeric representation. A discrete representation of data by numerals. [610.5]

numeric shift. A control for selecting the numeric character set on an alphanumeric keyboard or printer. *Contrast with*: **alphabetic shift.** *See also*: **shift character.** [610.5]

nybble (nibble). *See:* **quartet.** [610.1]

OBC. Abbreviation for **optical bar code.** *See:* **bar code.** [610.2]

object. (1) Pertaining to the outcome of an assembly or compilation process. *See also:* **object code; object module; object program.** [610.12]
(2) A program constant or variable. [610.12]
(3) An encapsulation of data and services that manipulate that data. *See also:* **object-oriented design.** [610.12]

object code. Computer instructions and data definitions in a form output by an assembler or compiler. An object program is made up of object code. *Contrast with:* **source code.** [610.12]

object extraction. *See:* **image segmentation.** [610.4]

object language. *See:* **target language.** [610.12]

object module. A computer program or subprogram that is the output of an assembler or compiler. *See also:* **load module; object program.** [610.12]

object-oriented design. A software development technique in which a system or component is expressed in terms of objects and connections between those objects. *See also:* **data structure-centered design; input-process-output; modular decomposition; rapid prototyping; stepwise refinement; structured design; transaction analysis; transform analysis.** [610.12]

object-oriented language. A programming language that allows the user to express a program in terms of objects and messages between those objects. Examples include Smalltalk and LOGO. [610.12]

object program. A computer program that is the output of an assembler or compiler. *Contrast with:* **source program.** *Syn:* **target program.** *See also:* **object module.** [610.12]

occupational title standard. A standard that describes the characteristics of the general areas of work or profession. [610.12]

occurrence. An individual instance of an entity, record, or item, containing a specific set of values for its constituent parts. [610.5]

OCR. Acronym for **optical character recognition.** [610.2]

OCR-A. Acronym for **optical character recognition-A.** [610.2]

OCR-B. Acronym for **optical character recognition-B.** [610.2]

octad (octade). A group of three bits used to represent one octal digit. [610.1]

octal. (1) Pertaining to a selection in which there are eight possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of eight. [610.1]

octal digit. A numeral used to represent one of the eight digits in the octal numeration system; 0, 1, 2, 3, 4, 5, 6, or 7. [610.1]

octal notation. Any notation that uses the octal digits and the radix 8. [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

octal number. (1) A quantity that is expressed using the octal numeration system. [610.1]
(2) Loosely, an octal numeral. [610.1]

octal number system.* *See:* **octal numeration system.** [610.1]
*Deprecated.

octal numeral. A numeral in the octal numeration system. For example, the octal numeral 14 is equivalent to the decimal numeral 12. [610.1]

octal numeration system. The numeration system that uses the octal digits and the radix 8. *Syn:* **octal system.** [610.1]

octal point. The radix point in the octal numeration system. [610.1]

octal system. *See:* **octal numeration system.** [610.1]

octal-to-binary conversion. The process of converting an octal numeral to an equivalent binary numeral. For example, octal 213.2 is converted to binary 10001011.01. [610.1]

octal-to-decimal conversion. The process of converting an octal numeral to an equivalent decimal numeral. For example, octal 213.2 is converted to decimal 139.25. [610.1]

octary tree. A tree of order 8. *Note*: Such a tree is typically used to store three-dimensional data. *Syn*: **octonary tree; octtree.** [610.5]

octet. A group of eight adjacent digits operated upon as a unit. [610.1, 610.5]

octodenary. (1) Pertaining to a selection in which there are 18 possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of 18. [610.1]

octonary tree. *See*: **octary tree.** [610.5]

octonary. *See:* **octal.** [610.1]
*Deprecated.

octtree. *See*: **octary tree.** [610.5]

odd parity. (1) An error detection method in which the number of ones in a binary word, byte, character, or message is maintained as an odd number. [610.1]
(2) The property possessed by a binary word, byte, character, or message that has an odd number of ones. [610.1]

odd-even check. *See:* **parity check.** [610.1]

odd-even sort. *See*: **Batcher's parallel sort.** [610.5]

office automation. The automation of information traffic through the use of any or all of the following: voice processing; word and data processing; reprographics; records processing and micrographics; telecommunications. *See also:* **automatic calendar; electronic mail; electronic office; paperless office.** [610.2]

office of the future. *See:* **electronic office.** [610.2]

off-line. Pertaining to a device or process that is not under the direct control of the central processing unit of a computer. *Contrast with:* **on-line (2).** [610.12]

offset. (1) The difference between the loaded origin and the assembled origin of a computer program. *Syn:* **relocation factor.** [610.12]
(2) A number that must be added to a relative address to determine the address of the storage location to be accessed. This number may be the difference defined in (1) or another number defined in the program. *See also:* **base address; indexed address; relative address; self-relative address.** [610.12]

omni-font character recognition. Character recognition of many or all character fonts. *Contrast with:* **single-font character recognition.** [610.2]

OMR. Acronym for **optical mark reading.** [610.2]

on-line. (1) Pertaining to a system or mode of operation in which input data enter the computer directly from the point of origin or output data are transmitted directly to the

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

point where they are used. For example, an airline reservation system. *Contrast with:* **batch.** *See also:* **conversational; interactive; real time.** [610.12]
(2) Pertaining to a device or process that is under the direct control of the central processing unit of a computer. [610.12] *Contrast with:* **off-line.** [610.12]

on-line compiler. *See:* **incremental compiler.** [610.12]

on-line dialog. *See:* **dialog.** [610.2]

on-line ordering. *See:* **teleordering.** [610.2]

one-address instruction. A computer instruction that contains one address field. For example, an instruction to load the contents of location A. *Syn:* **single-address instruction; single-operand instruction.** *Contrast with:* **multiaddress instruction; two-address instruction; three-address instruction; four-address instruction; zero-address instruction.** [610.12]

one-ahead addressing. A method of implied addressing in which the operands for a computer instruction are understood to be in the storage locations following the locations of the operands used for the last instruction executed. *Contrast with:* **repetitive addressing.** [610.12]

one-level address. *See:* **direct address.** [610.12]

one-plus-one address instruction. A computer instruction that contains two address fields, the second containing the address of the instruction to be executed next. For example, an instruction to load the contents of location A, then execute the instruction at location B. *Contrast with:* **two-plus-one address instruction; three-plus-one address instruction; four-plus-one address instruction.** [610.12]

one-question/one-response interaction. *See:* **question-and-answer interaction.** [610.2]

ones complement. The diminished-radix complement of a binary numeral, which is formed by subtracting each digit from 1. For example, the ones complement of 1101 is 0010. *Syn:* **complement on one; inverse; inverse binary state.** [610.1]

one-way chain. *See:* **linked list.** [610.5]

op code (opcode). *See:* **operation code.** [610.12]

open-address hashing. Hashing in which collision resolution is handled by inserting an item that has a duplicate hash value into another available position in the hash table. *See also:* **double hashing; linear probing; random probing; uniform probing.** *Contrast with:* **separate chaining.** [610.5]

open-loop control. Pertaining to a control system in which the output is permitted to vary in accordance with the inherent characteristics of the system, and no function of the output is used as feedback to the system. *Contrast with:* **closed-loop system.** [610.2]

open subroutine. A subroutine that is copied into a computer program at each place that it is called. *Syn:* **direct insert subroutine.** *Contrast with:* **closed subroutine.** *See also:* **inline code; macro.** [610.12]

operand. A variable, constant, or function upon which an operation is to be performed. For example, in the expression $A = B + 3$, B and 3 are the operands. [610.1, 610.12]

operating system. A collection of software, firmware, and hardware elements that controls the execution of computer programs and provides such services as computer resource allocation, job control, input/output control, and file management in a computer system. [610.12]

operation. (1) In computer mathematics, the action specified by an operator on one or more operands. For example, in the expression $A = B + 3$, the process of adding B to 3 to obtain A. [610.1a, 610.12]
(2) In programming, a defined action that can be performed by a computer system; for example, addition, comparison, branching. *Note:* Unlike the mathematical meaning, such an operation may not involve an operator or operands; for example, the operation Halt. [610.12]
(3) The process of running a computer system in its intended environment to perform its intended functions. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

operational. (1) Pertaining to a system or component that is ready for use in its intended environment. [610.12]
(2) Pertaining to a system or component that is installed in its intended environment. [610.12]
(3) Pertaining to the environment in which a system or component is intended to be used. [610.12]

operational character. *See:* **control character.** [610.5]

operational testing. Testing conducted to evaluate a system or component in its operational environment. *Contrast with:* **development testing.** *See also:* **acceptance testing; qualification testing.** [610.12]

operation and maintenance phase. The period of time in the software life cycle during which a software product is employed in its operational environment, monitored for satisfactory performance, and modified as necessary to correct problems or to respond to changing requirements. [610.12]

operation code. A character or set of characters that specifies a computer operation; for example, the code BNZ to designate the operation "branch if not zero." *Syn:* **op code.** [610.12]

operation exception. An exception that occurs when a program encounters an invalid operation code. *See also:* **addressing exception; data exception; overflow exception; protection exception; underflow exception.** [610.12]

operation field. The field of a computer instruction that specifies the operation to be performed. *Syn:* **function field; operation part.** *Contrast with:* **address field.** [610.12]

operation part. *See:* **operation field.** [610.12]

operations analysis. *See:* **operations research.** [610.2]

operations research (OR). The design of models for complex problems concerning the optimal allocation of available resources, and the application of mathematical methods for the solution of these problems. *Syn:* **operations analysis.** [610.2]

operation table. A table that describes an arithmetic or logical function by listing all possible combinations of input values and giving the output value that corresponds to each. *See also:* **truth table.** [610.1]

operator. (1) A mathematical or logical symbol that represents an action to be performed in an operation. For example, in the expression $A = B + 3$, + is the operator, representing addition. [610.1, 610.12]
(2) A person who operates a computer system. [610.12]

operator field. *See:* **operation field.** [610.12]

operator manual. A document that provides the information necessary to initiate and operate a system or component. Typically described are procedures for preparation, operation, monitoring, and recovery. *Note:* An operator manual is distinguished from a user manual when a distinction is made between those who operate a computer system (mounting tapes, etc.) and those who use the system for its intended purpose. *See also:* **diagnostic manual; installation manual; programmer manual; support manual; user manual.** [610.12]

optical bar code (OBC). *See:* **bar code.** [610.2]

optical character. *See:* **graphic character.** [610.2]

optical character recognition (OCR). The automatic recognition of graphic characters using light-sensitive devices such as optical mark readers. *Contrast with:* **magnetic ink character recognition.** [610.2]

optical character recognition-A (OCR-A). An international standard optical font used on documents intended to be read by optical character recognition. *Note:* This font is generally considered to present a less natural appearance to the eye than OCR-B. [610.2]

optical character recognition-B (OCR-B). An international standard optical font used on

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

documents intended to be read by optical character recognition. *Note:* This font is generally considered to present a more natural appearance to the eye than OCR-A. [610.2]

optical font. A character font used in optical character recognition. For example, hand-printed character font, OCR-A, or OCR-B. [610.2]

optical image. The result of projecting a scene onto a surface. For example, the image of a scene formed on film by a camera lens. [610.4]

optical mark reading (OMR). The use of pattern recognition techniques to identify graphite marks by automatic means. [610.2]

OR. A Boolean operator having the property that if P is a statement, Q is a statement, R is a statement,..., then the OR of P, Q, R,... is true if and only if at least one statement is true. *Note:* P OR Q is often represented by PVQ or P + Q. *Syn:* **Boolean add; inclusive OR; disjunction; false add; logic add; logical add; OR-ELSE; union.** *Contrast with:* **exclusive OR.** [610.1]

P	Q	PVQ
0	0	0
0	1	1
1	0	1
1	1	1

Fig 54
OR Truth Table

order. (1) To place items in an arrangement in accordance with a specified set of rules. *Note*: The arrangement need not be linear. *See also*: **sort (1).** [610.5]
(2) The result of an arrangement as in (1). [610.5]
(3) In a tree, the maximum number of subtrees of any node. [610.5]
(4) * *See*: **sequence (1).** [610.5]
* Deprecated.

order by merging. To order the items of a set by splitting the set into subsets, ordering the subsets, and merging the subsets. *See also*: **sequence by merging; sort by merging.** [610.5]

order clash. In software design, a type of structure clash in which a program must deal with two or more data sets that have been sorted in different orders. *See also:* **data structure-centered design.** [610.12]

ordered list. A list in which the data items are arranged in some specific order, either physically or logically by some key. *Contrast with*: **unordered list.** [610.5]

ordered tree. A tree in which the left-to-right order of the subtrees of a given node is significant. *Contrast with*: **unordered tree.** [610.5]

ordering bias. The manner and degree by which the order of a set of items departs from the order of a randomly distributed set of items. The ordering bias of a set is inversely proportional to the effort required to sort the set. [610.1a, 610.5]

ordinary binary. *See:* **binary.** [610.1]

OR-ELSE. *See:* **OR.** [610.1]

origin. The address of the initial storage location assigned to a computer program in main memory. *See also:* **assembled origin; loaded origin.** *Contrast with:* **starting address.** [610.12]

orphan prevention. The ability of a text formatter to avoid placing the final one or two lines of a paragraph at the top of a page. *See also:* **widow prevention.** [610.2]

OSAM. Acronym for **overflow sequential access method.** [610.5]

oscillating sort. An external merge sort in which sorts and merges are performed alternately; that is, the first two subsets are sorted and merged, the next subset is sorted and merged with the previously merged subsets, and so on, until all subsets are sorted and merged. [610.5]

outcome-oriented simulation. A simulation in which the end result is considered more important than the process by which it is obtained; for example, a simulation of a radar system that uses methods far different

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052888 2

from those used by the actual radar, but whose output is the same. *Contrast with:* **process-oriented simulation.** [610.3]

output. (1) Pertaining to data transmitted to an external destination. [610.12] (2) Pertaining to a device, process, or channel involved in transmitting data to an external destination. [610.12] (3) To transmit data to an external destination. [610.12] (4) Loosely, output data. [610.12] *Contrast with:* **input.** [610.12]

output assertion. A logical expression specifying one or more conditions that program outputs must satisfy in order for the program to be correct. *Contrast with:* **input assertion; loop assertion.** *See also:* **inductive assertion method.** [610.12]

output buffer. *See:* **buffer.** [610.5]

overflow. (1) The condition that arises when the result of an arithmetic operation exceeds the capacity of the number representation system used in a digital computer. [610.1] (2) The carry digit arising from this condition. *Syn:* arithmetic overflow. [610.1]

overflow area. A physical location in which data are placed when there is no available space in the primary data area. Overflow areas may be allocated within stored record, physical blocks, disk tracks, or disk cylinders. [610.5]

overflow error. The error caused by an overflow condition in computer arithmetic.[610.1]

overflow exception. An exception that occurs when the result of an arithmetic operation exceeds the size of the storage location designated to receive it. *See also:* **addressing exception; data exception; operation exception; protection exception; underflow exception.** [610.12]

overflow sequential access method (OSAM). An access method for handling data overflow from ISAM. [610.5]

overhead operation. *See:* **housekeeping operation.** [610.12]

overhead time. The amount of time a computer system spends performing tasks that do not contribute directly to the progress of any user task; for example, time spent tabulating computer resource usage for billing purposes. [610.12]

overlay. (1) A storage allocation technique in which computer program segments are loaded from auxiliary storage to main storage when needed, overwriting other segments not currently in use. [610.12] (2) A computer program segment that is maintained in auxiliary storage and loaded into main storage when needed, overwriting other segments not currently in use. [610.12] (3) To load a computer program segment from auxiliary storage to main storage in such a way that other segments of the program are overwritten. [610.12]

overlay supervisor. A routine that controls the sequencing and positioning of overlays. [610.12]

overload. To assign an operator, identifier, or literal more than one meaning, depending upon the data types associated with it at any given time during program execution. [610.12]

overprint. In text formatting, to print the same or different characters at the same position on an output page. Used to create bold-face type, underlining, and special characters. [610.2]

p. Abbreviation for **pico.** [610.1]

pack. To store data in a compact form in a storage medium, using known characteristics of the data and medium in such a way as to permit recovery of the data. *See also:* **packed data.** *Contrast with:* **unpack.** [610.5, 610.12]

package. A separately compilable software component consisting of related data types, data objects, and subprograms. *See also:* **data abstraction; encapsulation; information hiding.** [610.12]

packaging. In software development, the assignment of modules to segments to be

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

handled as distinct physical units for execution by a computer. [610.12]

packed array. An array in which all data elements in the set have non-trivial values. *Syn*: **dense list.** [610.5]

packed binary data. Binary data stored in a compact form in a storage medium, using known characteristics of the data and the medium to permit recovery of the data.[610.5]

packed data. Data stored in a compact form in a storage medium, using known characteristics of the data and the medium to permit recovery of the data. *See also*: **packed binary data; packed decimal data.** [610.5]

packed decimal data. Integer data stored in a compact form in a storage medium, using known characteristics of the data and the medium to permit recovery of the data. In the most common implementation, each decimal digit is represented in binary, occupying four bits, and the right-most decimal digit is followed by a four-bit sign digit (hexadecimal A,C,E, or F for positive; B or D for negative). *Syn*: **signed packed decimal data.** *See also*: **unsigned packed decimal data.** [610.5]

decimal	275_{10}
packed decimal	$0010\ 0111\ 0101\ 1111_2 = 275F_{16}$
decimal	-91_{10}
packed decimal	$0000\ 1001\ 0001\ 1011_2 = 091B_{16}$

pad. To fill an item such as a record or block with one or more filler characters in order to satisfy some prescribed condition. For example, in order to right justify a seven-character string in a ten-position field, three blank characters are used to **pad** the data. *See also*: **character fill; zero fill.** [610.5]

padding. (1) The technique of filling out a fixed-length block of data with dummy characters, words, or records. [610.12]
(2) Dummy characters, words, or records used to fill out a fixed-length block of data. [610.12]

page. (1) A fixed-length segment of data or of a computer program treated as a unit in storage allocation. *See also:* **paging.** [610.5, 610.12]
(2) In a virtual storage system, a fixed-length segment of data or of a computer program that has a virtual address and is transferred as a unit between main and auxiliary storage. [610.5, 610.12]
(3) A screenful of information on a video display terminal. [610.12]

page breakage. A portion of main storage that is unused when the last page of data or of a computer program does not fill the entire block of storage allocated to it. *See also:* **paging.** [610.12]

page eject character. *See*: **form feed character.** [610.5]

page frame. A block of main storage having the size of, and used to hold, a page. *See also:* **paging.** [610.12]

page makeup. *See:* **photocomposition; computer-aided page makeup.** [610.2]

pager. A routine that initiates and controls the transfer of pages between main and auxiliary storage. *See also:* **paging.** [610.12]

page swapping. The exchange of pages between main storage and auxiliary storage. *See also:* **paging.** [610.12]

page table. A table that identifies the location of pages in storage and gives significant attributes of those pages. *See also:* **paging.** [610.12]

page turning. *See:* **paging (3).** [610.12]

page zero. In the paging method of storage allocation, the first page in a series of pages. [610.12]

pagination. *See:* **automatic pagination.** [610.2]

paging. (1) A storage allocation technique in which programs or data are divided into fixed-length blocks called pages, main storage is divided into blocks of the same length called page frames, and pages are stored in page frames, not necessarily contiguously or in logical order. *Syn:* **block allocation.** *Contrast with:* **contiguous allocation.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

(2) A storage allocation technique in which programs or data are divided into fixed-length blocks called pages, main storage is divided into blocks of the same length called page frames, and pages are transferred between main and auxiliary storage as needed. *See also:* **anticipatory paging; demand paging; virtual storage.** [610.12]
(3) The transfer of pages as in (2). *Syn:* **page turning.** [610.12]
See also: **page; page breakage; page frame; page swapping; page table; page zero; pager; working set.** [610.12]

panning. *See:* **scrolling.** [610.2]

paperless office. An office that has been automated so that no paper documents are needed. *See also:* **electronic office; office automation.** [610.2]

paper throw character. *See:* **form feed character.** [610.5]

paper traffic. *See:* **information traffic.** [610.2]

parallel. (1) Pertaining to the simultaneous transfer, occurrence, or processing of the individual parts of a whole, such as the bits of a character, using separate facilities for the various parts. *Contrast with:* **serial (1).** [610.12]
(2) *See:* **concurrent.** [610.12]

parallel addition. Addition that is performed concurrently on all digit places of the operands. *Note:* This technique uses partial sums and partial carries to obtain its results. *Contrast with:* **serial addition.** [610.1]

parallel construct. A program construct consisting of two or more procedures that can occur simultaneously. [610.12]

parameter. (1) A variable that is given a constant value for a specified application. *See also:* **adaptation parameter.** [610.12]
(2) A constant, variable, or expression that is used to pass values between software modules. *See also:* **argument; formal parameter.** [610.12]

parenthesis-free notation. *See:* **prefix notation.** [610.1]

parent node. In a tree, a node having a given node as a child node. *Syn:* **father.** *Contrast with:* **child node; dependent node.** *See also:* **descendant node; logical parent; physical parent.** [610.5]

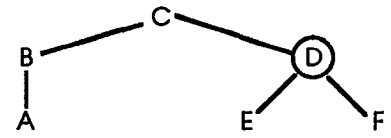


Fig 55
Node D Is the Parent Node for Node E

parent segment. In a hierarchical database, a segment that has one or more dependent segments, called child segments, below it in a hierarchy. [610.5]

parity. (1) An error detection method in which the total number of ones in a binary word, byte, character, or message is set to an odd or even number by appending a redundant bit. This number is subsequently checked to ensure that it remains odd or even. [610.1]
(2) The property of oddness or evenness possessed by a word, byte, character, or message. This property is determined by the total number of ones. *See also:* **even parity; odd parity.** [610.1]

parity bit. A binary digit appended to a binary word, byte, character, or message to make the total number of ones an odd or an even number. *See also:* **parity check.** [610.1]

parity check. A check to determine whether the total number of ones in a binary word, byte, character, or message is odd or even. *Syn:* **even-odd check; odd-even check.** [610.1]

parity error. The failure of a binary word, byte, character, or message to pass a parity check. [610.1]

parse. To determine the syntactic structure of a language unit by decomposing it into more elementary subunits and establishing the relationships among the subunits. For example, to decompose blocks into statements, statements into expressions, expressions into operators and operands. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

parser. A software tool that parses computer programs or other text, often as the first step of assembly, compilation, interpretation, or analysis. [610.12]

partial carry. (1) A carry process in which the carry digits are stored temporarily, instead of being processed as they occur. *Contrast with:* **complete carry.** *See also:* **cascaded carry; partial sum.** [610.1]
(2) The numeral that represents the carry digits generated in (1). [610.1]

partial correctness. In proof of correctness, a designation indicating that a program's output assertions follow logically from its input assertions and processing steps. *Contrast with:* **total correctness.** [610.12]

partially inverted file. A file that has been inverted on some of its secondary keys. *Contrast with:* **fully inverted file.** [610.5]

partial product. The result obtained by multiplying the multiplicand by one of the digits of the multiplier. *Syn:* **intermediate product.** [610.1]

partial sum. The result obtained from the addition of two or more numbers without regard to carries. *Note:* In the binary numeration system, the partial sum is the same result as is obtained from the exclusive-OR operation. *See also:* **cascaded carry.** [610.1]

partition exchange sort. *See:* **quicksort.** [610.5]

partition sort. *See:* **quicksort.** [610.5]

partitioned access. The process of storing and retrieving data from storage in such a way that the data is divided into subunits, called members, and the data may be processed as a whole or member by member. *Note:* The directory used to retrieve each member is stored along with the data. *See also:* **basic partitioned access method; partitioned data set.** [610.5]

partitioned data set (PDS). A file that is divided into subunits, called members, each of which may be processed individually. *Contrast with:* **indexed file; sequential file.** [610.5]

partitioning. Decomposition; the separation of the whole into its parts. [610.12]

parts program. A set of computer instructions used to control the action of a numerical control machine in producing a particular manufactured part. [610.2]

pass. A single cycle in the processing of a set of data, usually performing part of an overall process. For example, a pass of an assembler through a source program; a pass of a sort program through a set of data. [610.12]

pass/fail criteria. Decision rules used to determine whether a software item or a software feature passes or fails a test. *See also:* **test criteria.** [610.12]

passive data dictionary. A data dictionary that is only a repository for data definitions. *Note:* No active measures are taken to ensure that the data dictionary is consistent with the data items actually used in the system. *Syn:* **stand-alone data dictionary.** *Contrast with:* **active data dictionary.** [610.5]

patch. (1) A modification made directly to an object program without reassembling or recompiling from the source program. [610.12]
(2) A modification made to a source program as a last-minute fix or afterthought. [610.12]
(3) Any modification to a source or object program. [610.12]
(4) To perform a modification as in (1), (2), or (3). [610.12]

path. (1) In software engineering, a sequence of instructions that may be performed in the execution of a computer program. [610.12]
(2) In file access, a hierarchical sequence of directory and subdirectory names specifying the storage location of a file. [610.12]
(3) In the critical path method, any sequence of activities that goes from the beginning to the end of a project. [610.2]
(4) In a hierarchical database, a sequence of segments encountered in traversing from the root segment to an individual dependent segment. [610.5]
(5) With respect to a network or graph, some sequence of nodes such that each successive node is connected to its predecessor by an edge. *See also:* **simple path.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

path analysis. Analysis of a computer program to identify all possible paths through the program, to detect incomplete paths, or to discover portions of the program that are not on any path. [610.12]

path condition. A set of conditions that must be met in order for a particular program path to be executed. [610.12]

path expression. A logical expression indicating the input conditions that must be met in order for a particular program path to be executed. [610.12]

pathological coupling. A type of coupling in which one software module affects or depends upon the internal implementation of another. *Contrast with:* **common-environment coupling; content coupling; control coupling; data coupling; hybrid coupling.** [610.12]

path testing. Testing designed to execute all or selected paths through a computer program. *Contrast with:* **branch testing; statement testing.** [610.12]

pattern. A meaningful regularity that can be used to classify objects or other items of interest. [610.4]

pattern class. One of a set of mutually exclusive categories into which a pattern can be classified. *Syn:* **category; class.** [610.4]

pattern classification. The process of assigning patterns to pattern classes. *Syn:* **pattern identification.** [610.4]

pattern identification. *See:* **pattern classification.** [610.4]

pattern recognition. The analysis, description, identification, and classification of objects or other meaningful regularities by automatic or semiautomatic means. *Syn:* **machine recognition.** [610.4]

pattern segmentation. The process of determining which regions or areas of interest in an image or other set of data constitute patterns of interest for pattern recognition. [610.4]

pattern-sensitive fault. *See:* **data-sensitive fault.** [610.12]

pause. To suspend the execution of a computer program. *Syn:* **halt (2).** *Contrast with:* **stop.** [610.12]

PCA. Acronym for **physical configuration audit.** [610.12]

PCS. Acronym for **print contrast signal.** [610.2]

PDL. Acronym for **program design language.** [610.12]

PDR. Acronym for **preliminary design review.** [610.12]

PDS. Acronym for **partitioned data set.** [610.5]

peer graphics. *See:* **information graphics.** [610.2]

pel. *See:* **pixel.** [610.4]

perfective maintenance. Software maintenance performed to improve the performance, maintainability, or other attributes of a computer program. *Contrast with:* **adaptive maintenance; corrective maintenance.** [610.12]

performance. The degree to which a system or component accomplishes its designated functions within given constraints, such as speed, accuracy, or memory usage. [610.12]

performance requirement. A requirement that imposes conditions on a functional requirement; for example a requirement that specifies the speed, accuracy, or memory usage with which a given function must be performed. *Contrast with:* **design requirement; functional requirement; implementation requirement; interface requirement; physical requirement.** [610.12]

performance specification. A document that specifies the performance characteristics that a system or component must possess. These characteristics typically include speed, accuracy, and memory usage. Often part of a requirements specification. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

performance testing. Testing conducted to evaluate the compliance of a system or component with specified performance requirements. *See also:* **functional testing.** [610.12]

perimeter. In image processing, the number of pixels in the border of a region. [610.4]

period. The time interval between successive events in a discrete simulation. [610.3]

permutation. An ordered sequence of a given number of items chosen from a set. *Contrast with*: **combination.** [610.5]

permutation index. An automatic index in which each item appears repeatedly, each time with a different word of the item as the first word, followed by the subsequent words in the item, then by that part of the item that came before the word. *See also:* **keyword and context index; permutation on subject headings index.** [610.2]

permutation on subject headings (POSH) index. A permutation index in which the item entries are subject headings. [610.2]

personal computer. A single-user microcomputer designed for personally controllable applications. *See also:* **desktop computer; laptop computer; home computer.** [610.2]

personal computing. (1) Computing performed using a personal computer. [610.2]
(2) Computing performed in an environment in which the user has complete control over the data and access to software with which the data may be manipulated. *Syn:* **personal processing.** [610.2]

personal processing. *See:* **personal computing.** [610.2]

PERT. Acronym for **program evaluation and review technique.** [610.2]

Petri net. An abstract, formal model of information flow, showing static and dynamic properties of a system. A Petri net is usually represented as a graph having two types of nodes (called places and transitions) connected by arcs, and markings (called tokens) indicating dynamic properties. [610.3a, 610.12]

photocomposition. The formation of text and graphics into discrete camera-ready pages. *Syn:* **page makeup.** *See also:* **computer-aided page makeup.** [610.2]

phototypesetting. The preparation of textual material for printing using an optical system with a light source, a type store, a lens system, and a light-sensitive recording medium. *See also:* **computer-aided typesetting.** [610.2]

PHR. *See*: **physical record.** [610.5]

physical. Pertaining to the representation and storage of data on a data medium such as magnetic disk, or to characteristics of the data such as the length of data elements or records. *Contrast with*: **logical.** [610.5]

physical child segment. In a hierarchical database, a child segment in a physical database. *See also*: **logical child segment.** [610.5]

physical configuration audit (PCA). An audit conducted to verify that a configuration item, as built, conforms to the technical documentation that defines it. *See also:* **functional configuration audit.** [610.12]

physical database. (1) A database as it is actually stored. [610.5]
(2) A database containing a collection of related segments or records that are physically stored together. *Note*: Segments within a physical database are known as physical segments. *Contrast with*: **logical database.** [610.5]

physical data model. A data model that represents the implementation of the data contained in a data structure. *Contrast with*: **logical data model.** [610.5]

physical model. A model whose physical characteristics resemble the physical characteristics of the system being modeled; for example, a plastic or wooden replica of an airplane. *Contrast with:* **symbolic model.** *See also:* **iconic model; mock-up; scale model.** [610.3]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

physical parent segment. In a hierarchical database, a parent segment in a physical database. *See also*: **logical parent segment.** [610.5]

physical record (PHR, PR). (1) A record whose characteristics depend on the manner or form in which it is stored, retained, or moved. *Note*: A physical record may consist of all or part of a logical record or several physical records. [610.5]
(2) That which is accessed by a single read or write operation. [610.5]

physical requirement. A requirement that specifies a physical characteristic that a system or system component must possess; for example, material, shape, size, weight. *Contrast with:* **design requirement; functional requirement; implementation requirement; interface requirement; performance requirement.** [610.12]

physical segment. In a hierarchical database, the smallest unit of accessible data. *See also*: **physical child segment; physical parent segment; physical twin segment.** [610.5]

physical sequential access. *See*: **sequential access.** [610.5]

physical structure. The representation and storage of a database on a data medium. *See also*: **conceptual schema; reorganization.** [610.5]

physical twin segment. In a hierarchical database, a twin segment in a physical database. *Contrast with*: **logical twin segment.** [610.5]

physical volume. *See*: **volume.** [610.5]

pico (p). A prefix indicating 10^{-12}. [610.1]

pictorial pattern recognition. The recognition of patterns in visual or pictorial data. [610.4]

picture. *See:* **image.** [610.4]

picture data. Data that are associated with a picture specification. *Syn*: **pictured data.** *See also*: **binary picture data; decimal picture data.** [610.5]

pictured data. *See*: **picture data.** [610.5]

picture element. *See:* **pixel.** [610.4]

picture processing. *See:* **image processing.** [610.4]

picture specification. A character-by-character description of the composition and characteristics of the representation of some data item; for example, the picture S99V999 (S = sign character; 9 = decimal digit character; V = radix point character) may be used to describe the following items, resulting in the picture data as indicated: [610.5]

value	picture data
.06	+00.060
-10.342	-10.342
3	+03.000

pipeline. A software or hardware design technique in which the output of one process serves as input to a second, the output of the second process serves as input to a third, and so on, often with simultaneity within a single cycle time. [610.12]

pixel. In image processing, the smallest element of a digital image that can be assigned a gray level. *Note:* This term originated as a contraction for "picture element." *Syn:* **pel; picture element; resolution cell.** *See also:* **edge pixel; line pixel.** [610.4]

place. *See:* **digit place.** [610.1]

place value. In a positional notation system, the power of the radix that corresponds to a given place. For example, in a decimal integer the place values from right to left are 1, 100, etc. [610.1]

plan standard. A standard that describes the characteristics of a scheme for accomplishing defined objectives or work within specified resources. [610.12]

playback. *See:* **reversible execution.** [610.12]

pocket sort. *See*: **distribution sort.** [610.5]

point. *See:* **radix point.** [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

pointer. A data item that specifies the location of another data item; for example, a data item that specifies the address of the next employee record to be processed. *Syn:* **link.** *See also:* **pointer segment; stack pointer.** [610.5, 610.12]

pointer data. Data used to represent the addresses of other data items. [610.5]

pointer optimization. A database reorganization technique in which database access is made more efficient by reestablishing the pointers within the database so that fewer pointers are needed to represent the database structure. [610.5]

pointer segment. A segment in a database that establishes a parent/child relationship between segments. *Note:* The segment contains only a pointer to the physical child segment for its parent segment. [610.5]

point-of-sale terminal. A device for recording sales data in machine readable form at the time each sale is made. [610.2]

point operator. An image operator that assigns a gray level to each output pixel based on the gray level of the corresponding input pixel. *Contrast with:* **neighborhood operator.** [610.4]

Polish notation. *See:* **prefix notation.** [610.1]

polyphase merge sort. An unbalanced merge sort in which the distribution of the sorted subsets is based on a polynomial series such as the Fibonacci series. *See also:* **cascade merge sort.** [610.5]

pop. *See:* **pull.** [610.5]

populate. *See:* **load.** [610.5]

population. The number of records in a file or database. [610.5]

portability. The ease with which a system or component can be transferred from one hardware or software environment to another. *Syn:* **transportability.** *See also:* **machine independent.** [610.12]

portable computer. A personal computer that is designed and configured to permit transportation as a piece of handheld luggage. *Note:* Federal regulations limit use of the term "portable" to objects weighing no more than 21 pounds. *See also:* **laptop computer; transportable computer.** [610.2]

portrait image. *See:* **cine-oriented image.** [610.2]

port-to-port time. The elapsed time between the application of a stimulus to an input interface and the appearance of the response at an output interface. *See also:* **response time; think time; turnaround time.** [610.12]

POSH. Acronym for **permutation on subject headings.** *See:* **permutation on subject headings index.** [610.2]

position. *See:* **digit place.** [610.1]

positional notation. A number representation system that makes use of an ordered set of digits, such that the value contributed by each digit depends on its position as well as on the digit value. *Syn:* **positional representation.** *See also:* **binary numeration system; binary-coded-decimal system; biquinary numeration system; decimal numeration system; Gray code.** [610.1]

positional representation. *See:* **positional notation.** [610.1]

postamble breakpoint. *See:* **epilog breakpoint.** [610.12]

postfix notation. A method of forming mathematical expressions in which each operator is preceded by its operands. For example, A added to B and the result multiplied by C is expressed as AB + CX. *Syn:* **reverse Polish notation; suffix notation.** *Contrast with:* **infix notation; prefix notation.** [610.1]

postmortem dump. A dump that is produced upon abnormal termination of a computer program. *See also:* **change dump; dynamic dump; memory dump; selective dump; snapshot dump; static dump.** [610.12]

postorder traversal. The process of traversing a binary tree in a recursive fashion as fol-

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052896 1

lows: the left subtree is traversed, then the right tree is traversed, then the root is visited. *Contrast with*: **inorder traversal; preorder traversal.** *Syn*: **endorder traversal.** *See also*: **converse postorder traversal.**[610.5]

postprocessor. A computer program or routine that carries out some final processing step after the completion of the primary process; for example, a routine that reformats data for output. *Contrast with:* **preprocessor.** [610.12]

PR. *See*: **physical record.** [610.5]

practices. Requirements employed to prescribe a disciplined uniform approach to the software development process. *See also:* **conventions; standards.** [610.12]

pragma. *See:* **pseudo-instruction.** [610.12]

pre-tested iteration. *See:* **WHILE.** [610.12]

preamble breakpoint. *See:* **prolog breakpoint.** [610.12]

precision. The degree of exactness or discrimination with which a quantity is stated; for example, a precision of 2 decimal places versus a precision of 5 decimal places. *Contrast with:* **accuracy.** *See also*: **double precision; multiple precision; single precision; triple precision.** [610.1a, 610.5, 610.12]

precompiler. A computer program or routine that processes source code and generates equivalent code that is acceptable to a compiler. For example, a routine that converts structured FORTRAN to ANSI-standard FORTRAN. *See also:* **preprocessor.** [610.12]

predictive coding. An image compression technique that uses the gray levels of preceding pixels to predict the gray level of the current pixel, so that only the difference between the predicted and measured value needs to be encoded. [610.4]

predictive model. A model in which the values of future states can be predicted or are hypothesized; for example, a model that predicts weather patterns based on the current value of temperature, humidity, wind speed, and so on at various locations. [610.3]

prefix notation. A parenthesis-free method of forming mathematical expressions devised by the Polish logician Jan Lukasiewicz, in which each operator is immediately followed by its operands. For example, A added to B and the result multiplied by C is expressed as X+ABC. *Syn:* **Lukasiewicz notation; parenthesis-free notation; Polish notation.** *Contrast with:* **infix notation; postfix notation.** [610.1]

preliminary design. (1) The process of analyzing design alternatives and defining the architecture, components, interfaces, and timing and sizing estimates for a system or component. *See also:* **detailed design.** [610.12]
(2) The result of the process in (1). [610.12]

preliminary design review (PDR). (1) A review conducted to evaluate the progress, technical adequacy, and risk resolution of the selected design approach for one or more configuration items; to determine each design's compatibility with the requirements for the configuration item; to evaluate the degree of definition and assess the technical risk associated with the selected manufacturing methods and processes; to establish the existence and compatibility of the physical and functional interfaces among the configuration items and other items of equipment, facilities, software and personnel; and, as applicable, to evaluate the preliminary operational and support documents. *See also:* **critical design review; system design review.** [610.12]
(2) A review as in (1) of any hardware or software component. [610.12]

preorder traversal. The process of traversing a binary tree in a recursive fashion as follows: the root is visited, then the left subtree is traversed, then the right subtree is traversed. *Contrast with*: **inorder traversal; postorder traversal.** *See also*: **converse preorder traversal.** [610.5]

preprocessing. An operation performed before a primary process; for example, in pattern recognition, processing in which patterns are simplified to make classification easier. [610.4]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

preprocessor. A computer program or routine that carries out some processing step prior to the primary process; for example, a precompiler or other routine that reformats code or data for processing. *Contrast with:* **postprocessor.** [610.12]

prescriptive model. A model used to convey the required behavior or properties of a proposed system; for example, a scale model or written specification used to convey to a computer supplier the physical and performance characteristics of a required computer. *Contrast with:* **descriptive model.** [610.3]

presentation graphics. The use of a computer to produce high quality, high resolution graphical output. *Contrast with:* **information graphics.** [610.2]

prestore. To store data that are required by a computer program or routine before the program or routine is entered. [610.12]

prettyprinting. The use of indentation, blank lines, and other visual cues to show the logical structure of a program. [610.12]

preventive maintenance. Maintenance performed for the purpose of preventing problems before they occur. [610.12]

primary data element. A data element within a record that represents the subject of that record; for example, the data element "name" in a record containing "name," "city of birth," and "data of birth." *Contrast with:* **attribute data element.** [610.5]

primary key. (1) In sorting and searching, the key that is given the highest priority within a group of related keys. For example, after sorting, the values in the primary key will be in the given order, independent of the values of the other fields. *Syn:* **major key; prime key.** *Contrast with:* **secondary key.** [610.5]

(2) In a relation, a specific minimal set of attributes that functionally determines all other attributes in the relation, and thus uniquely differentiates one entity from another. *Note:* More than one set of attributes with this property may exist. Each such set is known as a candidate key, but only one is chosen as the primary key. *See also:* **alternate key; candidate key.** [610.5]

primary space allocation. The amount of space that is reserved for a particular file when it is initially defined. *Contrast with:* **secondary space allocation.** [610.5]

prime attribute. An attribute that forms all or part of the primary key of a relation. *Contrast with:* **nonprime attribute.** [610.5]

prime key. *See:* **primary key.** [610.5]

primitive Boolean function. A Boolean expression having the property that all other Boolean expressions can be constructed using it alone. [610.1]

primitive data structure. A data structure that can be directly operated upon by machine-level instructions. Examples include integer, real, character, logical, and pointer. *Contrast with:* **nonprimitive data structure.** [610.5]

primitive type. *See:* **atomic type.** [610.12]

print contrast ratio. In optical character recognition, the ratio obtained by subtracting the reflectance at an inspection area from the maximum reflectance found within a specified distance from that area, and dividing the result by that maximum reflectance. *Contrast with:* **print contrast signal.** [610.2]

print contrast signal (PCS). In optical character recognition, a measure of the contrast between a printed character and the paper on which the character is printed. *Contrast with:* **print contrast ratio.** [610.2]

print control character. A control character for print operations such as line spacing, page ejection, or carriage return. [610.5]

print data set. A data set in which data that is to be printed are stored. [610.5]

print formatter. *See:* **text formatter.** [610.2]

print formatting. *See:* **text formatting.** [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052898 5

print record. A record in a print data set.[610.5]

priority. The level of importance assigned to an item. [610.12]

priority interrupt. An interrupt performed to permit execution of a process that has a higher priority than the process currently executing. [610.12]

priority queue. A list to which items may be appended to or retrieved from any position, depending on some property of the item being added or removed. *Note*: This data structure is misnamed in that it contradicts the definition of queue. [610.5]

private type. A data type whose structure and possible values are defined but are not revealed to the user of the type. *See also:* **information hiding.** [610.12]

privileged instruction. A computer instruction that can be executed only by a supervisory program. [610.12]

privileged state. *See:* **supervisor state.** [610.12]

probabilistic model. *See:* **stochastic model.** [610.3]

probing. *See*: **linear probing; quadratic probing; random probing; uniform probing.** [610.5]

problem state. In the operation of a computer system, a state in which programs other than the supervisory program can execute. *Syn:* **slave state; user state.** *Contrast with:* **supervisor state.** [610.12]

problem-oriented language. A programming language designed for the solution of a given class of problems. Examples are list processing languages, information retrieval languages, simulation languages. [610.12]

procedural cohesion. A type of cohesion in which the tasks performed by a software module all contribute to a given program procedure, such as an iteration or decision process. *Contrast with:* **coincidental cohesion; communicational cohesion; functional cohesion; logical cohesion; sequential cohesion; temporal cohesion.** [610.12]

procedural language. A programming language in which the user states a specific set of instructions that the computer must perform in a given sequence. All widely-used programming languages are of this type. *Syn:* **procedure-oriented language.** *Contrast with:* **nonprocedural language.** *See also:* **algebraic language; algorithmic language; list processing language; logic programming language.** [610.12]

procedure. (1) A course of action to be taken to perform a given task. [610.12]
(2) A written description of a course of action as in (1); for example, a documented test procedure. [610.12]
(3) A portion of a computer program that is named and that performs a specific action. [610.12]

procedure-oriented language. *See:* **procedural language.** [610.12]

process. (1) A sequence of steps performed for a given purpose; for example, the software development process. [610.12]
(2) An executable unit managed by an operating system scheduler. *See also:* **task; job.** [610.12]
(3) To perform operations on data. [610.12]

process control. Automatic control in which a computer is used to regulate continuous operations such as chemical processes, military operations, or manufacturing operations. *See also:* **numerical control.** [610.2]

processing cycle. A single, complete execution of data processing that is periodically repeated. *Syn:* **data processing cycle.** *See also:* **annual cycle; daily cycle; monthly cycle; quarterly cycle; weekly cycle.** [610.2]

process management. The direction, control, and coordination or work performed to develop a product or perform a service. Example is quality assurance. [610.12]

process model. A model of the processes performed by a system; for example, a model that represents the software development

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

process as a sequence of phases. *Contrast with:* **structural model.** [610.3]

process-oriented simulation. A simulation in which the process is considered more important than the outcome; for example, a model of a radar system in which the objective is to replicate exactly the radar's operation, and duplication of its results is a lesser concern. *Contrast with:* **outcome-oriented simulation.** [610.3]

process standard. A standard that deals with the series of actions or operations used in making or achieving a product. [610.12]

product. (1) The result of a multiplication operation. [610.1]
(2) A relational operator that builds a relation from two specified relations consisting of all possible concatenated pairs of tuples, one from each of the two original relations. *See also*: **difference; intersection; join; projection; selection; union.** [610.5]

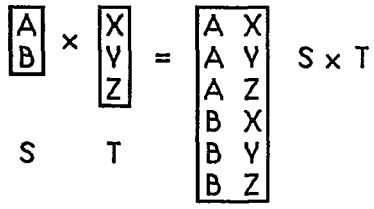


Fig 56
Product

product analysis. The process of evaluating a product by manual or automated means to determine if the product has certain characteristics. [610.12]

product baseline. In configuration management, the initial approved technical documentation (including, for software, the source code listing) defining a configuration item during the production, operation, maintenance, and logistic support of its life cycle. *Contrast with:* **allocated baseline; developmental configuration; functional baseline.** *See also:* **product configuration identification.** [610.12]

product code. *See:* **bar code; universal product code.** [610.2]

product configuration identification. The current approved or conditionally approved technical documentation defining a configuration item during the production, operation, maintenance, and logistic support phases of its life cycle. It prescribes all necessary physical or form, fit and function characteristics of a configuration item, the selected functional characteristics designated for production acceptance testing, and the production acceptance tests. *Contrast with:* **allocated configuration identification; functional configuration identification.** *See also:* **product baseline.** [610.12]

product engineering. The technical processes to define, design, and construct or assemble a product. [610.12]

product management. The definition, coordination, and control of the characteristics of a product during its development cycle. Example is configuration management. [610.12]

product specification. (1) A document that specifies the design that production copies of a system or component must implement. *Note:* For software, this document describes the as-built version of the software. *See also:* **design description.** [610.12]
(2) A document that describes the characteristics of a planned or existing product for consideration by potential customers or users. [610.12]

product standard. A standard that defines what constitutes completeness and acceptability of items that are used or produced, formally or informally, during the software engineering process. [610.12]

product support. The providing of information, assistance, and training to install and make software operational in its intended environment and to distribute improved capabilities to users. [610.12]

production library. A software library containing software approved for current operational use. *Contrast with:* **master library; software development library; software repository; system library.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

professional standard. A standard that identifies a profession as a discipline and distinguishes it from other professions. [610.12]

program. (1) *See:* **computer program.** [610.12] (2) To write a computer program. [610.12]

program counter. *See:* **instruction counter.** [610.12]

program data set. A data set in which user programs are stored. [610.5]

program definition language. *See:* **program design language.** [610.12]

program design language (PDL). A specification language with special constructs and, sometimes, verification protocols, used to develop, analyze, and document a program design. *See also:* **hardware design language; pseudo code.** [610.12]

program editor. A text editor user to enter, alter, and view source code for computer programs. Such an editor may have features that make it sensitive to the syntax of the source language on which it operates. *Contrast with:* **document editor.** [610.2]

program evaluation and review technique (PERT). A variation of the critical path method in which minimum, maximum, and most likely times are used to estimate the mean and standard deviation of each activity item; these values are used to compute estimated path times and to find the critical path; and the critical path values are used to find the standard deviation of the completion time for the whole project. [610.2]

program flowchart (flow chart). *See:* **flowchart.** [610.12]

program instruction. A computer instruction in a source program. *Note:* A program instruction is distinguished from a computer instruction that results from assembly, compilation, or other interpretation process. [610.12]

program library. *See:* **software library.** [610.12]

program listing. A printout or other human readable display of the source and, sometimes, object statements that make up a computer program. [610.12]

programmable breakpoint. A breakpoint that automatically invokes a previously specified debugging process when initiated. *See also:* **code breakpoint; data breakpoint; dynamic breakpoint; epilog breakpoint; prolog breakpoint; static breakpoint.** [610.12]

programmed instruction. A self-instructional method using materials that lead the student through a systematic sequence of steps to a predetermined learning objective. [610.2]

programmer manual. A document that provides the information necessary to develop or modify software for a given computer system. Typically described are the equipment configuration, operational characteristics, programming features, input/output features, and compilation or assembly features of the computer system. *See also:* **diagnostic manual; installation manual; operator manual; support manual; user manual.** [610.12]

programming language. A language used to express computer programs. *See also:* **assembly language; high-order language; machine language.** *Contrast with:* **query language; specification language.** [610.12]

programming support environment. An integrated collection of software tools accessed via a single command language to provide programming support capabilities throughout the software life cycle. The environment typically includes tools for specifying, designing, editing, compiling, loading, testing, configuration management, and project management. Sometimes called integrated programming support environment. *See also:* **scaffolding.** [610.12]

programming system. A set of programming languages and the support software (editors, compilers, linkers, etc.) necessary for using these languages with a given computer system. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

program mutation. (1) A computer program that has been purposely altered from the intended version to evaluate the ability of test cases to detect the alteration. *See also:* **mutation testing.** [610.12]
(2) The process of creating an altered program as in (1). [610.12]

program network chart. A diagram that shows the relationship between two or more computer programs. [610.12]

program-sensitive fault. A fault that causes a failure when some particular sequence of program steps is executed. *Contrast with:* **data-sensitive fault.** [610.12]

program status word (PSW). (1) A computer word that contains information specifying the current status of a computer program. The information may include error indicators, the address of the next instruction to be executed, currently enabled interrupts, and so on. [610.12]
(2) A special-purpose register that contains a program status word as in (1). [610.12]
Syn: **status word.** [610.12]

program structure diagram. *See:* **structure chart.** [610.12]

program support library. *See:* **software development library.** [610.12]

program synthesis. The use of software tools to aid in the transformation of a program specification into a program that realizes that specification. [610.12]

project. *See:* **projection.** [610.5]

project evaluation and review technique.* *See:* **program evaluation and review technique.** [610.2]
* Deprecated.

project file. A central repository of material pertinent to a project. Contents typically include memos, plans, technical reports, and related items. *Syn:* **project notebook.** [610.12]

projection. A relational operator that extracts specified attributes from a relation and results in a relation containing only those attributes. *See also:* **difference; intersection; join; product; selection; union.** *Syn:* **project.** [610.5]

Name	Homeroom
Mary	26A
Joe	43
Harry	27
Michael	25
Susan	25
Mickey	41

Fig 57
Projection of Relation *Students* in Fig 11 on Attributes NAME and HOMEROOM

projection/join normal form (PJ/NF or 5NF). *See:* **fifth normal form.** [610.5]

project library. *See:* **software development library.** [610.12]

project notebook. *See:* **project file.** [610.12]

project plan. A document that describes the technical and management approach to be followed for a project. The plan typically describes the work to be done, the resources required, the methods to be used, the procedures to be followed, the schedules to be met, and the way that the project will be organized. For example, a software development plan. [610.12]

prolog breakpoint. A breakpoint that is initiated upon entry into a program or routine. *Syn:* **preamble breakpoint.** *Contrast with:* **epilog breakpoint.** *See also:* **code breakpoint; data breakpoint; dynamic breakpoint; programmable breakpoint; static breakpoint.** [610.12]

prompt. (1) A symbol or message displayed by a computer system, requesting input from the user of the system. [610.12]
(2) To display a symbol or message as in (1). [610.12]

proof of correctness. (1) A formal technique used to prove mathematically that a computer program satisfies its specified requirements. *See also:* **assertion; formal specification; inductive assertion method; partial correctness; total correctness.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

(2) A proof that results from applying the technique in (1). [610.12]

propagated error. An error that occurs in a GIVEN operation and is passed along to a later operation. *Contrast with:* **inherited error.** [610.1]

propagation sort. *See:* **bubble sort.** [610.5]

proportional spacing. Text formatting and output that takes into account the width of each character, rather than allocating the same amount of horizontal space to characters of all widths. [610.2]

protection character. A character used to replace a suppressed zero in order to avoid error or false statements; for example, in the string "$********50.03" the asterisk is the protection character. [610.5]

protection exception. An exception that occurs when a program attempts to write into a protected area in storage. *See also:* **addressing exception; data exception; operation exception; overflow exception; underflow exception.** [610.12]

protocol. A set of conventions that govern the interaction of processes, devices, and other components within a system. [610.12]

prototype. A preliminary type, form, or instance of a system that serves as a model for later stages or for the final, complete version of the system. [610.3, 610.12]

prototyping. A hardware and software development technique in which a preliminary version of part or all of the hardware or software is developed to permit user feedback, determine feasibility, or investigate timing or other issues in support of the development process. *See also:* **rapid prototyping.** [610.12]

pseudo code (pseudocode). A combination of programming language constructs and natural language used to express a computer program design. For example: [610.12]

```
IF the data arrives faster than expected,
  THEN reject every third input.
  ELSE process all data received.
ENDIF
```
[610.12]

pseudo instruction. A source language instruction that provides information or direction to the assembler or compiler and is not translated into a target language instruction. For example, an instruction specifying the desired format of source code listings. *Syn:* **pragma; pseudo-op; pseudo operation.** [610.12]

pseudo-op. *See:* **pseudo instruction.** [610.12]

pseudo operation. *See:* **pseudo instruction.** [610.12]

pseudo-random number sequence. A sequence of numbers, determined by some defined arithmetic process, that is sufficiently close to a random number sequence to permit its use in calculations formally requiring a random number sequence. [610.1]

pseudo-random. Pertaining to the approximation of true, statistical randomness. [610.1]

pseudo-random number. Any member of a sequence of numbers sufficiently close to a random number sequence to permit its use in calculations formally requiring random numbers. [610.1]

PSW. Acronym for **program status word.** [610.12]

pull. To retrieve data from a stack. *Contrast with:* **push.** *Syn:* **pop.** [610.5]

pure alphabetic. Pertaining to data that contains only the letters of the alphabet (AaBbCcDdEeFfGgHh...). *Contrast with:* **pure alphanumeric; pure numeric.** [610.5]

pure alphanumeric. Pertaining to data that contains only the letters of the alphabet (AaBbCcDdEeFfGgHh...) and the numerals (1234567890). *Contrast with:* **pure alphabetic; pure numeric.** [610.5]

pure binary numeration system. *See:* **binary numeration system.** [610.1]

pure binary. *See:* **binary.** [610.1]

pure numeric. Pertaining to data that contains only the numerals (1234567890). *Contrast with:* **pure alphabetic; pure alphanumeric.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

purged. Pertaining to a record that has been physically deleted from a file. *See also:* **active; inactive; logically deleted.** [610.2]

push. To append data onto a stack. *Contrast with*: **pull.** [610.5]

push-down list. *See*: **stack.** [610.5]

push-down stack. *See*: **stack.** [610.5]

push-down storage. *See*: **stack.** [610.5]

push-up list. *See*: **queue.** [610.5]

push-up storage. *See*: **queue.** [610.5]

put. To place an item into a set of items as in inserting a record into a file, or in representing a numerical value as a series of decimal digits. *Contrast with*: **get.** [610.5]

QA. Acronym for **quality assurance.** [610.12]

QC. Acronym for **quality control.** [610.12]

QISAM. Acronym for **queued indexed sequential access method.** [610.5]

QSAM. Acronym for **queued sequential access method.** [610.5]

quadary tree. A tree of order 4. *Note*: Such a tree is usually used to store two-dimensional data. *Syn*: **quaternary tree; quadtree.** [610.5]

quadratic probing. Open-address hashing in which collision resolution is handled by selecting positions at quadratically increasing increments from the original position in the hash table until an available position is found. *Contrast with*: **linear probing; random probing; uniform probing.** [610.5]

quadratic programming. In operations research, a particular type of nonlinear programming in which the function to be maximized or minimized is a quadratic function and the constraints are linear functions. [610.2]

quadratic selection sort. A tree selection sort in which the items in the set to be sorted are treated as nodes on a 4-ary tree. [610.5]

quadtree. *See*: **quadary tree.** [610.5]

qualification. The process of determining whether a system or component is suitable for operational use. [610.12]

qualification condition. In a query, a boolean condition that specifies which items in a database are to be accessed; that is, the qualification condition must be true for an item to be accessed by the query. *See also*: **atomic condition; item condition; query condition; record condition.** [610.5]

qualification testing. Testing conducted to determine whether a system or component is suitable for operational use. *See also:* **acceptance testing; development testing; operational testing.** [610.12]

qualified name. A name used to identify a component of a structure given by a sequence of names connected by periods; for example, in the following structure, the qualified name STUDENT(5).COURSE(1).INSTRUCTOR identifies the instructor for the first course taken by the fifth student. [610.5]

```
01 STUDENT(100),
   02 STUDENT_NUMBER      CHAR(5),
   02 COURSES(10),
      03 COURSE_NAME      CHAR(20),
      03 INSTRUCTOR       CHAR(20),
      03 GRADE            CHAR(2)
```

Fig 58
Qualified Name

quality. (1) The degree to which a system, component, or process meets specified requirements. [610.12]
(2) The degree to which a system, component, or process meets customer or user needs or expectations. [610.12]

quality assurance (QA). (1) A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements. [610.12]
(2) A set of activities designed to evaluate the process by which products are developed or manufactured. *Contrast with:* **quality control (1).** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052904 7

quality attribute. A feature or characteristic that affects an item's quality. *Syn:* **quality factor.** *Note:* In a hierarchy of quality attributes, higher level attributes may be called quality factors, lower level attributes called quality attributes. [610.12]

quality control (QC). *Note:* This term has no standardized meaning in software engineering at this time. Candidate definitions are: (1) A set of activities designed to evaluate the quality of developed or manufactured products. *Contrast with:* **quality assurance (2).** [610.12]
(2) The process of verifying one's own work or that of a co-worker. [610.12]
(3) Synonym for **quality assurance.** [610.12]

quality factor. *See:* **quality attribute.** *Note:* In a hierarchy of quality attributes, higher level attributes may be called quality factors, lower level attributes called quality attributes. [610.12]

quality metric. (1) A quantitative measure of the degree to which an item possesses a given quality attribute. [610.12]
(2) A function whose inputs are software data and whose output is a single numerical value that can be interpreted as the degree to which the software possesses a given quality attribute. [610.12]

quantitative data element. A data element containing numerical values that indicate quantity or amount. [610.5]

quantization. In image processing, a process in which each pixel in an image is assigned one of a finite set of gray levels. *See also:* **equal interval quantizing; equal probability quantizing.** [610.4]

quantize. To subdivide the range of values of a variable into a finite number of non-overlapping intervals, each of which is represented by an assigned value within the interval. For example, to represent a person's age as a number of whole years. [610.1]

quarter-end processing. The operations required to complete a quarterly cycle. [610.2]

quarterly cycle. One complete execution of a data processing function that must be performed once a quarter. *See also:* **annual cycle; daily cycle; monthly cycle; weekly cycle.** [610.2]

quartet. A group of four adjacent digits operated upon as a unit. *Syn:* **four-bit byte, nibble; nybble.** [610.1, 610.5]

quaterdenary. (1) Pertaining to a selection in which there are 14 possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of 14. [610.1]

quaternary. (1) Pertaining to a selection in which there are four possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of four. [610.1]

quaternary tree. *See:* **quadary tree.** [610.5]

query. A request to access information stored in a database; for example, "Print the names of all personnel with salary > 9000 and sex = F." *Note:* A query consists of an operation (Print) and a qualification condition (salary > 9000 and sex = F). *See also:* **query language.** [610.5]

query condition. A conjunction of two or more record conditions; for example: [610.5]
(LASTNAME = JONES and SEX = F) or (LASTNAME = GREEN and SEX = M). [610.5]

query language. A language used to access information stored in a database. *Contrast with:* **programming language; specification language.** *Syn:* **end-user language; search language.** *See also:* **query; relational language; report writer.** [610.5, 610.12]

question-and-answer interaction. An instruction method employed by some computer-assisted instruction systems, in which the student is asked to provide responses to questions asked by the system. Types of questions include short answer, multiple choice, matching, and drill-and-practice. *See also:* **dialog.** *Contrast with:* **instructional game; simulation.** *Syn:* **one-question/one-response interaction.** [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

queue. (1) A list in which items are appended to the last position in the list and retrieved from the first position in the list. That is, the next item to be retrieved is the item that has been in the list for the longest time. *Syn*: **push-up list; push-up storage.** *Contrast with*: **stack.** *See also*: **double-ended queue; priority queue.** [610.5, 610.12a]
(2) A line formed by items waiting for service in a system in which the next item to exit the line is the item that has been in the line for the longest time. [610.5]
(3) To arrange in, or to form a queue as in (1). [610.5]
(4) In queueing theory, a set of zero or more entities waiting to be serviced by a service facility. [610.3]

queued access method. An access method in which the transfer of data is synchronized between the processing unit and the storage device through the use of queues containing data waiting to be processed. *See also*: **queued indexed sequential access method; queued sequential access method.** *Contrast with*: **basic access method; direct access method.** [610.5]

queued indexed sequential access method (QISAM). A variation of the queued access method that allows indexed sequential access to the data. *See also*: **basic indexed sequential access method.** [610.5]

queued sequential access method (QSAM). A variation of the queued access method that allows sequential access to the data. *See also*: **basic sequential access method.** [610.5]

queueing model. A model consisting of service facilities and entities waiting in queues to be served; for example, a model depicting teller windows and customers at a bank. [610.3]

queueing network model. A model in which a process is described as a network in which each node represents a service facility rendering a given type of service and a queue for holding entities waiting to be served; for example, a model depicting a network of shipping routes and docking facilities at which ships must form queues in order to unload their cargo. [610.3]

queueing theory. The study of queues and the performance of systems that service entities that are organized into queues. *See also:* **queueing model; queueing network model.** [610.3]

quibinary code. A BCD code in which each decimal digit is represented by a seven-digit binary numeral. [610.1]

quicksort. An exchange sort in which the set of items to be sorted is divided into two subsets such that all items in one subset precede all items in the other subset according to the sort criteria, the two subsets are similarly partitioned, and the process continues until all items are in the correct order. *Syn*: **partition exchange sort; partition sort.** [610.5]

quiescing. The process of bringing a device or system to a halt by rejecting new requests for work. [610.12]

quinary. (1) Pertaining to a selection in which there are five possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of five. [610.1]

quindenary. (1) Pertaining to a selection in which there are 15 possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of 15. [610.1]

quintet. A group of five adjacent digits operated upon as a unit. *Syn*: **five-bit byte.** [610.1, 610.5]

quoted string. A character string that is enclosed by quotation marks or apostrophes. For example, "This is a quoted string." [610.5]

quotient. The result of a division operation. [610.1]

radix. A quantity whose successive integer powers are the implicit multipliers of the sequence of digits that represent a number in some positional notation systems. For example, if the radix is 5, then 143.2 means 1 times 5 to the second power, plus 4 times 5 to the first power, plus 3 times 5 to the zero power, plus 2 times 5 to the minus-one power. *Syn:* **base; base number; radix number.** [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

radix alignment. In text formatting, the formatting of numbers in a column such that their radix points, whether explicit or implicit, form a vertical line. *See also:* **decimal alignment.** [610.2]

radix complement. The complement obtained by subtracting each digit of a given numeral from the largest digit in the numeration system, then adding 1 to the least significant digit of the result and executing any required carries. For example, twos complement in binary notation, tens complement in decimal notation. *Syn:* **base complement; complement on *n*; noughts complement; true complement; zero complement.** *Contrast with:* **diminished-radix complement.** [610.1]

radix exchange sort. A radix sort in which items are compared and, if necessary, exchanged in multiple passes, using successive digits within the numeric representation of the sort key, starting with the most significant digit. *Syn:* **divide-and-conquer sort.** [610.5]

radix insertion sort. A radix sort in which each item is inserted into its proper position in the sorted set according to the digital properties of the numerical representation of the sort keys. [610.5]

radix list sort. A radix sort implemented using the list sorting technique. [610.5]

radix-minus-one complement. *See:* **diminished-radix complement.** [610.1]

radix notation. A positional representation system in which the ratio of the place values of adjacent digits is a positive integer (the radix). *Syn:* **radix numeration system; radix scale.** [610.1]

radix number. *See:* **radix.** [610.1]

radix numeration system. *See:* **radix notation.** [610.1]

radix point. In positional notation, the character, expressed or implied, that separates the integral part of a numerical expression from the fractional part. For example, binary point, decimal point, hexadecimal point, or octal point. *Syn:* **arithmetic point; base point; point.** [610.1]

radix point character. A character within a picture specification that represents the radix point. *Syn:* **virtual point picture character.** [610.5]

radix scale. *See:* **radix notation.** [610.1]

radix search. A searching technique that takes advantage of the digital properties of the numerical representation of the search keys. *See also:* **binary radix trie search; digital tree search; multiway radix trie search; radix trie search.** *Contrast with:* **radix sort.** [610.5]

radix sort. A sort that takes advantage of the digital properties of the numerical representation of the sort keys; for example, sorting on keys with base 10 representation by first sorting on the hundreds place, then the tens place, then the ones place. *See also:* **digital sort; radix exchange sort; radix insertion sort; radix list sort; straight radix sort.** *Contrast with:* **radix search.** [610.5]

radix transformation function. In hashing, a hash function the result of which is the original key in a different numerical base from its original base. For example, in the function below, the original key (assumed to be in base 10) is expressed in base 16.

Original key	Calculation	Hash value
72	$72_{10} = 48_{16}$	48
157	$157_{10} = 9D_{16}$	9D

[610.5]

radix trie search. A radix search in which the items in the set to be searched are placed in a trie. *Note:* The trie is traversed taking branches according to the search argument until a terminal node is encountered, and if the search is successful, the external node is equal to the search argument. *See also:* **binary radix trie search; multiway radix trie search.** [610.5]

ragged left margin. In text formatting, a left margin that is not aligned. *Contrast with:* **left justification.** [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

ragged right margin. In text formatting, a right margin that is not aligned. *Contrast with:* **right justification.** [610.2]

random. Pertaining to a process or variable whose outcome or value depends on chance or on a process that simulates chance, often with the implication that all possible outcomes or values have an equal probability of occurrence; for example, the outcome of flipping a coin or executing a computer-programmed random number generator.[610.3]

random access. (1) An access mode in which specific logical records are obtained from or placed into a file in a nonsequential manner. *Contrast with*: **direct access; sequential access.** [610.5]
(2) * *See*: direct access. [610.5]
* Deprecated.

random access method. * *See*: **direct access method.** [610.5]
* Deprecated.

random failure. A failure whose occurrence is unpredictable except in a probabilistic or statistical sense. *See also:* **intermittent fault; transient error.** [610.12]

randomizing. *See*: **hashing.** [610.5]

random number. A number selected by chance from a given set of numbers, and satisfying one or more of the standard tests for statistical randomness. [610.1]

random number sequence. (1) A sequence of random numbers, each of which is statistically independent of its predecessors. [610.1]
(2) Loosely, a pseudo-random number sequence. [610.1]
(3) A sequence of numbers in which no number can be predicted from knowledge of its predecessors. [610.5]

random-ordered list. *See*: **unordered list.** [610.5]

random probing. Open-address hashing in which collision resolution is handled by randomly selecting positions in the hash table until an available position is found. *Contrast with*: **linear probing; quadratic probing; uniform probing.** [610.5]

range check. A consistency check that ensures that an item of data falls between preestablished maximum and minimum values. [610.5]

rapid prototyping. A type of prototyping in which emphasis is placed on developing prototypes early in the development process to permit early feedback and analysis in support of the development process. *Contrast with:* **waterfall model.** *See also:* **data structure-centered design; incremental development; input-process-output; modular decomposition; object-oriented design; spiral model; stepwise refinement; structured design; transaction analysis; transform analysis.** [610.12]

rational number. A real number that can be expressed as a fraction x/y where x and y are integers and y is not equal to zero. *Contrast with*: **irrational number.** [610.5]

raw data. Data that has not been processed or reduced from its original form. [610.5]

read. To access data from a storage device or data medium. *Contrast with*: **write.** *See also*: **delete; destructive read; dirty read; nondestructive read; retrieve; update.** [610.5, 610.12]

read-back check. *See*: **echo check.** [610.5]

read-only access. A type of access to data in which the data may be read but not changed or deleted. *Contrast with*: **read/write access.** *See also*: **delete access; update access; write access.** *Syn*: **fixed.** [610.5]

read/write access. A type of access to data in which the data may be both retrieved, changed, and stored. *Contrast with*: **read-only access.** *See also*: **delete access; update access; write access.** [610.5]

real address. The address of a storage location in the main storage part of a virtual storage system. *Contrast with:* **virtual address.** [610.12]

real data. Data used to represent real numbers. *See also*: **binary coded decimal real data; floating-point real data.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052908 4

real fixed binary data. *See*: **fixed-point binary data.** [610.5]

real fixed decimal data. *See*: **fixed-point real data.** [610.5]

real float binary data. *See*: **floating-point data.** [610.5]

real float decimal data. *See*: **floating-point data.** [610.5]

real number. A member of the set of all positive and negative numbers, including integers, zero, mixed, fractional, rational, and irrational numbers. [610.1, 610.5]

real storage. The main storage portion of a virtual storage system. *Contrast with:* **virtual storage.** [610.12]

real time. (1) Pertaining to a system or mode of operation in which computation is performed during the actual time that an external process occurs, in order that the computation results can be used to control, monitor, or respond in a timely manner to the external process. *Contrast with:* **batch.** *See also:* **conversational; interactive; interrupt; on-line.** [610.12]
(2) In modeling and simulation, simulated time with the property that a given period of actual time represents the same period of time in the system being modeled; for example, in a simulation of a radar system, running the simulation for one second may result in the model advancing time by one second; that is, simulated time advances at the same rate as actual time. *Contrast with:* **fast time; slow time.** [610.3]

real type. A data type whose members can assume real numbers as values and can be operated on by real number arithmetic operations, such as addition, subtraction, multiplication, division, and square root. *Contrast with:* **character type; enumeration type; integer type; logical type.** [610.12]

real variable. A variable that may assume only real-number values. [610.1]

realm. *See*: area. [610.5]

recomplementation. The process of taking the complement of a complement. *Note:* The complement of a complement is the original numeral. [610.1]

reconstruction. *See:* **image reconstruction.** [610.4]

record. A set of data items, called fields, treated as a unit. For example, in stock control, the data for each invoice could constitute one record. *See also*: **database record; entity.** *Syn*: **data record.** [610.5, 610.12]

record condition. A conjunction of two or more item conditions such that the name of the data item in each condition is distinct. For example, "LASTNAME = 'JONES' and SEX = 'FEMALE.'" *See also*: **query condition.** [610.5]

record gap. * *See*: **interblock gap.** [610.5]
* Deprecated.

record layout. The arrangement and structure of data in a record. [610.5]

record length. The number of words or characters in a record. [610.5]

record length type. The category to which a record belongs by virtue of having fixed or variable length. [610.5]

record segmentation. The allocation of individual data items in a record to separate physical storage areas or to different physical devices. [610.5]

record type. The category to which a record belongs by virtue its format, content, or characteristics. [610.5]

record-locking. *See*: **lock.** [610.5]

recording area. In micrographics, the maximum useful area of a microfilm or other medium that can record information, including the image as well as the document marks. [610.2]

records processing. The process of manipulating, storing, and retrieving records in electronic form. *See also:* **office automation.** [610.2]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

recovery. (1) The restoration of a system, program, database, or other system resource to a prior state following a failure or externally caused disaster; for example, the restoration of a database to a point at which processing can be resumed following a system failure. *See also*: **backword recovery; forward recovery; inline recovery; rollback; rollforward.** [610.5]
(2) The restoration of a system, program, database, or other system resource to a state in which it can perform required functions. *See also:* **checkpoint.** [610.12a]

recursion. (1) A process in which a software module calls itself. *See also:* **simultaneous recursion.** [610.12]
(2) The process of defining or generating a process or data structure in terms of itself. [610.12]

recursive. (1) Pertaining to a software module that calls itself. [610.12]
(2) Pertaining to a process or data structure that is defined or generated in terms of itself. [610.12]

recursive data structure. A data structure that is defined in terms of itself. [610.5]

recursively defined sequence. A sequence in which each item after the first is determined using a given operation for which one or more of the operands include one or more of the preceding items. [610.5]

redefinition. (1) The process of changing a database schema by adding, removing, or renaming attributes or relations. [610.5]
(2) In a relation, the process of changing the data type or size of an attribute, or altering the characteristics of a domain. [610.5]

redundancy. In fault tolerance, the presence of auxiliary components in a system to perform the same or similar functions as other elements for the purpose of preventing or recovering from failures. *See also:* **active redundancy; diversity; homogeneous redundancy; standby redundancy.** [610.12]

reenterable. *See:* **reentrant.** [610.12]

reentrant. Pertaining to a software module that can be entered as part of one process while also in execution as part of another process and still achieve the desired results. *Syn:* **reenterable.** [610.12]

reentry point. The place in a software module at which the module is reentered following a call to another module. [610.12]

reference edge. *See:* **document reference edge.** [610.2]

reflected binary code. *See:* **Gray code.** [610.1]

reflected binary unit-distance code. *See:* **Gray code.** [610.1]

reflected code. *See:* **Gray code.** [610.1]

reformatting. *See*: **reorganization.** [610.5]

region. A connected subset of an image. [610.4]

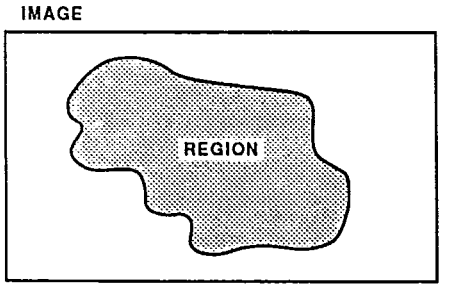


Fig 59
Illustration of Region

region growing. An image segmentation technique in which regions are formed by repeatedly taking the union of subregions that are similar in gray levels or textures. *See also:* **region partitioning.** [610.4]

region partitioning. An image segmentation technique in which regions are formed by repeatedly taking the union of subregions that are similar in gray levels or textures and by repeatedly splitting apart subregions that are dissimilar. *See also:* **region growing.** [610.4]

registered images. Two or more images of the same scene that have been positioned with respect to one another so that corresponding points in the images represent the same point in the scene. [610.4]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

registration. *See:* **image registration.** [610.4]

regression testing. Selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements. [610.12]

regular binary. *See:* **binary.** [610.1]

rehashing. *See*: **collision resolution.** [610.5]

relation. In a relational data model or relational database, a set of tuples, each of which has the same attributes. *Note*: Often thought of as a table of data. *See also*: **relation scheme.** [610.5]

relational algebra. An algebra that includes a set of relational operators, such as join and projection, to manipulate relations and the axioms of those operators. [610.5]

relational data model. (1) A data model whose pattern or organization is based on a set of relations, each of which consists of an unordered set of tuples. [610.5]
(2) A data model that provides for the expression of relationships among data elements as formal mathematical relations. [610.5]

relational database model. An external data model that represents a relational database. [610.5]

relational database schema. A collection of relation schemas that define the structural properties of a relational database. [610.5]

relational database. A database in which data are organized into one or more relations that may be manipulated using a relational algebra. *Contrast with*: **hierarchical database; network database.** [610.5]

relational file. (1) A file, consisting of tuples, in which all data items are associated via the same relationship. *Note:* Also called a flat file. [610.5]
(2) Any file resulting from relational algebra. [610.5]

relational language. A query language that may be used to access and retrieve data from a relational database. [610.5]

relational model. *See*: **relational data model; relational database model.** [610.5]

relational operator. An operator that performs an operation on relations; for example, the join or projection operators. *See also*: **relational algebra.** [610.5]

relationally complete. Pertaining to a query language or system that can be used to form expressions from a relational algebra.[610.5]

relation schema. The set of all attribute names for a relation. [610.5]

relationship. A directed connection between two or more data items or attributes. [610.5]

relative address. An address that must be adjusted by the addition of an offset to determine the address of the storage location to be accessed. *Contrast with:* **absolute address.** *See also:* **base address; indexed address; self-relative address.** [610.12]

relative error. The ratio of an error to the correct value. *Contrast with:* **absolute error.** [610.1]

relative loader. *See:* **relocating loader.**[610.12]

reliability. The ability of a system or component to perform its required functions under stated conditions for a specified period of time. *See also:* **availability; MTBF.** [610.12]

reliability growth. The improvement in reliability that results from correction of faults. [610.12]

reliability model. A model used to estimate, measure, or predict the reliability of a system; for example, a model of a computer system, used to estimate the total down time that will be experienced. [610.3]

relocatable. Pertaining to code that can be loaded into any part of main memory. The starting address is established by the loader, which then adjusts the addresses in the code

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

to reflect the storage locations into which the code has been loaded. *See also:* **relocating loader.** [610.12]

relocatable address. An address that is to be adjusted by the loader when the computer program containing the address is loaded into memory. *Contrast with:* **absolute address.** [610.12]

relocatable code. Code containing addresses that are to be adjusted by the loader to reflect the storage locations into which the code is loaded. *Contrast with:* **absolute code.** [610.12]

relocate. To move machine code from one portion of main memory to another and to adjust the addresses so that the code can be executed in its new location. [610.12]

relocating assembler. An assembler that produces relocatable code. *Contrast with:* **absolute assembler.** [610.12]

relocating loader. A loader that reads relocatable code into main memory and adjusts the addresses in the code to reflect the storage locations into which the code has been loaded. *Syn:* **relative loader.** *Contrast with:* **absolute loader.** [610.12]

relocation dictionary. The part of an object module or load module that identifies the addresses that must be adjusted when a relocation occurs. [610.12]

relocation factor. *See:* **offset (1).** [610.12]

remote batch entry. *See:* **remote job entry.** [610.12]

remote job entry (RJE). Submission of jobs through a remote input device connected to a computer through a data link. *Syn:* **remote batch entry.** [610.12]

remote-access data processing. Data processing in which some or all of the input-output functions are performed at locations away from the primary computer, connected to the primary computer by telecommunication facilities. [610.2]

reorder. * *See:* **order.** [610.5]
* Deprecated.

reorganization. (1) The process of rearranging the contents of a database so that space allocation is minimized and efficiency is maximized. Techniques include pointer optimization and garbage collection. *Syn:* **restructuring.** *See also:* **concurrent reorganization.** [610.5]
(2) The process of rearranging the logical schema or physical structure of a database. *Syn:* **reformatting.** [610.5]

repagination. *See:* **automatic pagination.** [610.2]

repeatability. *See:* **test repeatability.** [610.12]

repeated selection sort. A selection sort in which the set of items to be sorted is divided into subsets; one item that fits specified criteria is selected from each subset, forming a second-level subset; a selection sort is then applied to this second-level subset; the selected item is appended to the sorted set and is replaced in the second-level subset by the next eligible item in the original subset; and the process is repeated until all items are in the sorted set. *See also:* **tournament sort.** [610.5]

repeating field. A field within a record that may have multiple occurrences within a record; for example, the data element "Student Name" may have up to 30 occurrences within the following record structure: [610.5]

01	Course Name	20 characters
01	Instructor Name	25 characters
01	Students (30)	
	02 Student Name	25 characters
	02 Student Number	9 characters

Fig 60
Repeating Field

repeating group. A collection of data elements that may have multiple occurrences within a record; for example, the data elements representing the name and age of each dependent within an employee record.[610.5]

repetitive addressing. A method of implied addressing in which the operation field of an computer instruction is understood to

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052912 6

address the operands of the last instruction executed. *Contrast with:* **one-ahead addressing.** [610.12]

replay. *See:* **reversible execution.** [610.12]

report standard. A standard that describes the characteristics of describing results of engineering and management activities. [610.12]

report writer. A query language that can produce formatted reports using data from a database or other files. [610.5]

representation standard. A standard that describes the characteristics of portraying aspects of an engineering or management product. [610.12]

representational model. *See:* **descriptive model.** [610.3]

reproduce. *See*: **duplicate.** [610.5]

reprographics. Automated composition, production, and reproduction of printed material. Methods include photocomposition, computer-aided typesetting, and offset printing. *See also:* **office automation.** [610.2]

required hyphen. In word processing, a hyphen that is to appear in a word or phrase regardless of whether the word or phrase is divided to achieve justification; for example, the hyphen in "computer-aided design." *Note:* A required hyphen is not subject to hyphen drop. *Syn:* **embedded hyphen.** *Contrast with*: **discretionary hyphen.** [610.2]

requirement standard. A standard that describes the characteristics of a requirements specification. [610.12]

requirement. (1) A condition or capability needed by a user to solve a problem or achieve an objective. [610.12]
(2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents. [610.12]
(3) A documented representation of a condition or capability as in (1) or (2). [610.12]
See also: **design requirement; functional requirement; implementation requirement; interface requirement; performance requirement; physical requirement.** [610.12]

requirements analysis. (1) The process of studying user needs to arrive at a definition of system, hardware, or software requirements. [610.12]
(2) The process of studying and refining system, hardware, or software requirements. [610.12]

requirements phase. The period of time in the software life cycle during which the requirements for a software product are defined and documented. [610.12]

requirements review. A process or meeting during which the requirements for a system, hardware item, or software item are presented to project personnel, managers, users, customers, or other interested parties for comment or approval. Types include system requirements review, software requirements review. *Contrast with:* **code review; design review; formal qualification review; test readiness review.** [610.12]

requirements specification language. A specification language with special constructs and, sometimes, verification protocols, used to develop, analyze, and document hardware or software requirements. *See also:* **design language.** [610.12]

requirements specification. A document that specifies the requirements for a system or component. Typically included are functional requirements, performance requirements, interface requirements, design requirements, and development standards. *Contrast with:* **design description.** *See also:* **functional specification; performance specification.** [610.12]

rescue point. *See:* **restart point.** [610.12]

reserved word. A word in a programming language whose meaning is fixed by the rules of that language and which, in certain or all contexts, cannot be used by the programmer for any purpose other than its intended one. Examples include IF, THEN, WHILE. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

reset. To set a variable, register, or other storage location back to a prescribed state. *See also:* **clear; initialize.** [610.12]

resident control program. *See:* **kernel (1).** [610.12]

residual control. A microprogramming technique in which the meaning of a field in a microinstruction depends on the value in an auxiliary register. *Contrast with:* **bit steering.** *See also:* **two-level encoding.** [610.12]

residual error. The difference between an optimum result derived from experience or experiment and a theoretically exact result. [610.1]

resolution. (1) In image processing, the degree to which closely spaced objects in an image can be distinguished from one another. [610.4]
(2) In micrographics, the ability of a photographic system to record fine detail. [610.2]

resolution cell. *See:* **pixel.** [610.4]

resolution test chart. In micrographics, a chart containing a number of increasingly smaller horizontal and vertical lines of specific size and spacing, used to measure resolution. *See also:* **target.** [610.2]

resource allocation. *See:* **computer resource allocation.** [610.12]

resource management. The identification, estimation, allocation, and monitoring of the means used to develop a product or perform a service. Example is estimating. [610.12]

response time. The elapsed time between the end of an inquiry or command to an interactive computer system and the beginning of the system's response. *See also:* **port-to-port time; think time; turnaround time.** [610.12]

restart. To cause a computer program to resume execution after a failure, using status and results recorded at a checkpoint. [610.12]

restart point. A point in a computer program at which execution can be restarted following a failure. *Syn:* **rescue point.** [610.12]

restoration. *See:* **image restoration.** [610.4]

restore. To recover the state of a system, computer program, or database to a specific point. *See also:* **rollback; rollforward.** [610.5]

restructuring. *See:* **reorganization.** [610.5]

retirement. (1) Permanent removal of a system or component from its operational environment. [610.12]
(2) Removal of support from an operational system or component. [610.12]
See also: **software life cycle; system life cycle.** [610.12]

retirement phase. The period of time in the software life cycle during which support for a software product is terminated. [610.12]

retrieval code. In micrographics, a code used for manual or automatic retrieval of microimages. [610.2]

retrieve. To move data out of a storage device or data medium. *Contrast with:* **store (2).** *See also:* **read.** [610.5]

retrospective trace. A trace produced from historical data recorded during the execution of a computer program. *Note:* This differs from an ordinary trace, which is produced cumulatively during program execution. *See also:* **execution trace; subroutine trace; symbolic trace; variable trace.** [610.12]

return. (1) To transfer control from a software module to the module that called it. *See also:* **return code.** [610.12]
(2) To assign a value to a parameter that is accessible by a calling module; for example, to assign the value 25 to parameter AGE for use by a calling module. *See also:* **return value.** [610.12]
(3) A computer instruction or process that performs the transfer in (1). [610.12]

return code. A code used to influence the execution of a calling module following a return from a called module. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

return value. A value assigned to a parameter by a called module for access by the calling module. [610.12]

reusability. The degree to which a software module or other work product can be used in more than one computer program or software system. *See also:* **generality.** [610.12]

reusable. Pertaining to a software module or other work product that can be used in more than one computer program or software system. [610.12]

reverse execution. *See:* **reversible execution.** [610.12]

reverse leading. In photocomposition, the ability of some phototypesetting equipment to allow reverse movement of the photographic medium. *Note:* This technique permits the setting of side-by-side columns of text on the composed page. *See also:* **leading.** [610.2]

reverse Polish notation. *See:* **postfix notation.** [610.1]

reverse scrolling. In word processing, the process of moving the text across the display screen in the reverse direction from the normal reading direction. *See also:* **scrolling.** [610.2]

reversible execution. A debugging technique in which a history of program execution is recorded and then replayed under the user's control, in either the forward or backward direction. *Syn:* **backward execution; playback; replay; reverse execution.** [610.12]

review. A process or meeting during which a work product, or set of work products, is presented to project personnel, managers, users, customers, or other interested parties for comment or approval. Types include code review, design review, formal qualification review, requirements review, test readiness review. [610.12]

right justification. In text formatting, justification of text such that the right margin is aligned. *Contrast with:* **ragged right margin.** [610.2]

right-threaded tree. A threaded tree in which the right link field in each terminal node is made to point to its successors with respect to a particular order of traversal. *Contrast with:* **left-threaded tree.** [610.5]

ring. *See:* **circularly-linked list.** [610.5]

ring shift. *See:* **circular shift.** [610.1]

RJE. Acronym for **remote job entry.** [610.12]

robustness. The degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions. *See also:* **error tolerance; fault tolerance.** [610.12]

rollback. Backward recovery of a database in which recently applied changes to the current version of a database are reversed. *Note:* A journal or checkpoint file is used to determine which changes must be reversed. *Syn:* **backout.** *Contrast with:* **rollforward.** [610.5]

rollforward. Forward recovery of a database in which all or part of a database is restored using data from a backup or snapshot of the database. Changes since the backup are reapplied to the database to restore it to some recently existing state. *Contrast with:* **rollback.** [610.5]

roll in. To transfer data or computer program segments from auxiliary storage to main storage. *Contrast with:* **roll out.** *See also:* **swap.** [610.12]

roll out. To transfer data or computer program segments from main storage to auxiliary storage for the purpose of freeing main storage for other uses. *Contrast with:* **roll in.** *See also:* **swap.** [610.12]

root. *See:* **root node.** [610.5]

root compiler. A compiler whose output is a machine independent, intermediate-level representation of a program. A root compiler, when combined with a code generator, comprises a full compiler. [610.12]

rooted tree. *See:* **tree.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

root node. In a tree, the single node that is not a member of any subtree. *Note*: All other nodes are descendent nodes of the root node. *Syn*: **root**. *See also*: **nonterminal node; terminal node.** [610.5]

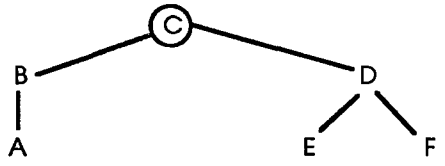


Fig 61
Node C Is the Root Node

root segment. A segment that is the root node in a database. *See also*: **database record; parent; path.** *Contrast with*: **child segment; dependent segment.** [610.5]

rotate. *See:* **circular shift.** [610.1]

round. To delete or omit one or more of the least significant digits in a representation of a number and to adjust the remaining digits according to some specified rule. *See also*: **round down; round off; round up.** *Contrast with*: **truncate.** [610.1a, 610.5]

round down. To round a number, making no adjustment to the numeral that is retained. For example, the decimal numeral 5.6789, when rounded down to two decimal places, becomes 5.67. *Syn:* **truncate (1).**[610.1, 610.5a]

rounding error. The error introduced by rounding a number. *Syn:* **round-off error.** [610.1]

round off. (1) To round, adjusting the part of the numeral that is retained by rounding down any digit less than 5, rounding up any digit greater than 5, and rounding 5 up or down to the even digit. For example, 5.5 would be rounded off to 6, and 4.5 rounds off to 4. [610.5]
(2) To round, adjusting the part of the numeral that is retained by rounding down any digit less than 5, rounding up any digit equal or greater than 5. For example, 5.5 rounds off to 6, 4.5 rounds off to 5. [610.5]
(3) *See:* **round.** [610.1]

round-off error. *See:* **rounding error.** [610.1]

round up. To round a number, adjusting the numeral that is retained by adding 1 to its least significant digit and executing any carries required. For example, the decimal numeral 5.6789 when rounded up to two decimal places becomes 5.68. [610.1, 610.5a]

routine. A subprogram that is called by other programs and subprograms. *Note:* The terms "routine," "subprogram," and "subroutine" are defined and used differently in different programming languages; the preceding definition is advanced as a proposed standard. *See also:* **coroutine; subroutine.** [610.12]

row. (1) *See*: **tuple.** [610.5]
(2) A horizontally corresponding set of entries in a table. *See also*: **tuple.** *Contrast with*: **column.** [610.5]

row binary. Pertaining to the binary representation of data in which adjacent positions in a row correspond to adjacent binary digits. For example, each row in an 80-column card may be used to represent 80 consecutive bits of a binary word. *Contrast with:* **column binary.** [610.1]

row-major order. A method for storing the elements of a matrix in computer memory, in which the elements are ordered in a row-by-row manner—that is, all elements of row 1, followed by all elements of row 2, etc. *Contrast with*: **column-major order.** [610.5]

row vector. A matrix with only one row. That is, a matrix of size 1-by-*n*. *Contrast with*: **column vector.** [610.5]

rub-out character. *See*: **delete character.** [610.5]

rule-based language. A nonprocedural language that permits the user to state a set of rules and to express queries or problems that use these rules. *See also:* **declarative language; interactive language.** [610.12]

run. (1) In software engineering, a single, usually continuous, execution of a computer program. *See also:* **run time.** [610.12]
(2) To execute a computer program. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052916 3

(3) In image processing, a sequence of consecutive pixels that all have the same gray level. [610.4]
(4) In sorting, two or more successive items in a set that are in the proper order according to the specified sorting criteria. [610.5]

run length. The number of pixels in a run. [610.4]

run length encoding. An image compression technique in which the rows of an image are represented as sequences of runs, each with a given run length and gray level. [610.4]

running footer. In text formatting, a line of text that is automatically placed at the bottom of each page of a document. *Contrast with:* **running header.** *Syn:* **footer.** [610.2]

running header. In text formatting, a line of text that is automatically placed at the top of each page of a document. *Contrast with:* **running footer.** *Syn:* **header.** [610.2]

running time. *See:* **execution time.** [610.12]

run stream. *See:* **job stream.** [610.12]

run time. (1) The instant at which a computer program begins to execute. [610.12]
(2) The period of time during which a computer program is executing. [610.12]
(3) *See:* **execution time.** [610.12]

SAM. Acronym for **sequential access method.** [610.5]

sampling. In image processing, the technique of dividing an image into disjoint regions, selecting a single point in each region to represent the region, and measuring the brightness or color of each of these points. [610.4]

scaffolding. Computer programs and data files built to support software development and testing, but not intended to be included in the final product. For example, dummy routines or files, test case generators, software monitors, stubs. *See also:* **programming support environment.** [610.12]

scalar. A data item used to represent a single number or entity. *Contrast with:* **vector.** [610.1a, 610.5]

scale. (1) To adjust the representation of a quantity so that its value is brought within a specified range. [610.5]
(2) The difference between the original and resulting adjustment as in (1). [610.5]
(3) A system of mathematical notation such as fixed-point or floating point. [610.5]
(4) To multiply the representation of a number by a factor in order to bring its range within prescribed limits. [610.1]

scale factor. A number used as a factor in a scaling operation. *See also:* **scale.** [610.1]

scale model. A physical model that resembles a given system, with only a change in scale; for example, a replica of an airplane one tenth the size of the actual airplane. [610.3]

scan. To examine a set of items sequentially. [610.5]

scatter storage. *See:* **hashing.** [610.5]

scenario. A set of initial conditions and a sequence of events used to develop, test, or apply a system, model, or simulation. [610.3]

scheduler. A computer program, usually part of an operating system, that schedules, initiates, and terminates jobs. [610.12]

schema. A description of the logical structure of a database. *See also:* **data model; physical schema.** [610.5]

schema definition language. *See:* **data definition language.** [610.5]

schema language. *See:* **data definition language.** [610.5]

scientific notation. A notation system in which a number is expressed as a coefficient multiplied by a power of ten. [610.1]

SCN. Acronym for **specification change notice.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

scratch. (1) To physically erase data from its medium. [610.5]
(2) To logically delete the identification of data from its medium. [610.5]

scratch file. A file used as a work area to hold data temporarily. [610.5]

scratchpad area (SPA). A portion of computer memory shared by a set of computer programs or processes for some special purpose. For example, memory used by two programs for interprocess communication. [610.5]

screen editor. *See:* **full-screen editor.** [610.2]

scrolling. In word processing, the process of moving text across a display screen to create the effect of a viewing window moving on a large page of a document. An operator may scroll left, right, up, or down in a document. *See also:* **reverse scrolling.** [610.2]

SDD. (1) Acronym for **software design description.** [610.12]
(2) (DoD) Acronym for software design document. *See:* **software design description.** [610.12]

SDP. Acronym for **software development plan.** [610.12]

SDR. Acronym for **system design review.** [610.12]

SDS. Acronym for **sequential data set.** [610.5]

search. (1) The examination of a set of items to find all those having a desired property or properties. For example, to find all items in a file that meet some search criterion. [610.5]
(2) To examine a set of items as in (1). [610.5]
(3) To retrieve the results of an examination as in (1). [610.5]
(4) To retrieve the first item within a set of items as in (1). [610.5]

search argument. In a search, the value compared with the search key of each item in the set being searched. *See also*: **condition.** [610.5]

search criterion. In a search, the relationship that a search key must have to the search argument in order for the search to be successful. For example, "NAME equals 'SMITH;'" "SALARY greater than 10000." [610.5]

search cycle. That portion of a search that is repeated for each item in the set being searched. [610.5]

search key. In a search, the key within each item in the set being searched that is compared to the search argument. *Syn*: **seek key.** [610.5]

search language. * *See*: **query language.** [610.5]
* Deprecated.

search length. (1) For a node in a search tree, the number of nodes that must be examined in order to find that node. [610.5]
(2) For a search tree, the average search length as in (1) for all nodes in the tree. [610.5]

search tree. (1) A tree into which items in a set are placed in order for the set to be searched. The tree is traversed according to some searching algorithm, making key comparisons until the search argument is found or the algorithm is halted. For example, a B-tree. [610.5]
(2) A multiways tree of order m in which each nonterminal node may contain $(m - 1)$ key values and each terminal node, called a leaf, contains associated data for one of the key values contained in its parent node. Each subtree is used to contain all the items with key values falling in the intervals formed by the key values contained in its root node. *See also*: B-tree; binary search tree; digital search tree. [610.5]

secondary access method. A collection of techniques designed to allow efficient access to all the target data or data records associated with a set of stated secondary key values in a query. [610.5]

secondary index. (1) A list associated with an inverted file in which entries in the list

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

point to records in the file that contain identical values for the key field on which the file is inverted. [610.5]
(2) In a hierarchical database, an index used to establish access to a physical or logical segment by a path different from the one provided by the primary key within the root segment. *Note*: A secondary index allows access on the basis of any field within the segment or any of its dependent segments with secondary indices. *See also*: **secondary processing sequence; source segment.** [610.5]

secondary key. (1) In sorting and searching, a key that is given lower priority than the primary key within a group of related keys. That is, after sorting, all items having the same primary key will be in order by the secondary key or keys. *Syn*: **minor key.** *Contrast with*: **primary key.** [610.5]
(2) Within a record, a key that is used to index that record but which does not necessarily uniquely identify that record. [610.5]

secondary processing sequence. In a hierarchical database, the hierarchical order of segment types in a physical or logical database resulting from a secondary index. [610.5]

secondary space allocation. The amount of space that is reserved for a particular file after the primary space allocation has been exhausted. *Note*: Some systems allow multiple secondary space allocation operations. When a secondary space allocation is granted to a particular file, that file is said to "increase its extents." *Contrast with*: **primary space allocation.** [610.5]

second generation language (2GL). *See:* **assembly language.** [610.12]

second normal form (2NF). One of the forms used to characterize relations; a relation is said to be in second normal form if it is in first normal form and if every nonprime attribute is fully functionally dependent on each candidate key of the relation. [610.5]

FIRST NORMAL FORM

ORDER1 = {ORDER-NO} + DATE + CUSTOMER-NO
+ CUSTOMER-NAME + CUSTOMER-ADDRESS
+ TOTAL-ORDER-AMOUNT
ITEM1 = {ORDER-NO + SEQUENCE-NO} + ITEM-NO
+ ITEM-DESCRIPTION + QUANTITY-ORDERED
+ UNIT-PRICE + EXTENDED-PRICE

SECOND NORMAL FORM

ORDER2 = {ORDER-NO} + DATE + CUSTOMER-NO
+ CUSTOMER-NAME + CUSTOMER-ADDRESS
+ TOTAL-ORDER-AMOUNT
ORDER-ITEM2 = {ORDER-NO + ITEM-NO}
+ QUANTITY-ORDERED
+ EXTENDED-PRICE
ITEM2 = {ITEM-NO} + ITEM-DESCRIPTION
+ UNIT-PRICE

Note: In first normal form, nonprime attributes ITEM-DESCRIPTION and UNIT-PRICE are not functionally dependent on candidate key SEQUENCE-NO. Keys shown in brackets.

Fig 62
Second Normal Form

security kernel. A small, self-contained collection of key security-related statements that works as a privileged part of an operating system, specifying and enforcing criteria that must be met for programs and data to be accessed. [610.12]

seek. * (1) *See*: **search.** [610.5]
(2) * *See*: **search cycle.** [610.5]
* Deprecated.

seek key. *See*: **search key.** [610.5]

segment. (1) One of the subsystems or combinations of subsystems that make up an overall system; for example, the accounts payable segment of a financial system. [610.12]
(2) In storage allocation, a self-contained portion of a computer program that can be executed without maintaining the entire program in main storage. *See also:* **page.** [610.12]
(3) A collection of data that is stored or transferred as a unit. [610.5, 610.12]
(4) In path analysis, a sequence of computer program statements between two consecutive branch points. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

(5) To divide a system, computer program, or data file into segments as in (1), (2), or (3). [610.12]
(6) A fixed-length unit of data that contains one or more data items. [610.5]
(7) In some databases, the smallest unit of data that can be retrieved or stored. *Syn*: **database segment.** *See also*: **child segment; dependent segment; logical segment; parent segment; physical segment; root segment; twin segment.** [610.5]

segmentation. *See:* **image segmentation.** [610.4]

select. (1) To identify, within a set of items, all items that meet a particular criterion. *See also*: **extract.** [610.5]
(2) *See*: **selection** (2). [610.5]

selection. (1) The process of identifying, within a set of items, all items that meet a particular criterion. [610.5]
(2) A relational operator that extracts specified tuples from a relation and results in a relation containing only those tuples. Also called **select.** *See also*: **difference; intersection; join; product; projection; union.** [610.5]

Student No.	Name	Grade	Homeroom
15	Mary	4	26A
21	Harry	4	27

Fig 63
Selection of Relation *Students* in Fig 11 where GRADE is ≤ 5

selection sort. A sort in which the items in a set are examined to find an item that fits a specified criterion; for example, the smallest item; this item is appended to the sorted set and removed from further consideration; and the process is repeated until all items are in the sorted set. *Syn*: **straight selection sort.** *See also*: **heapsort; quadratic selection sort; repeated selection sort; tree selection sort.** [610.5]

selective choice construct. *See:* **branch.** [610.12]

selective dump. A dump of designated storage location areas only. *See also:* **change dump; dynamic dump; memory dump; postmortem dump; snapshot dump; static dump.** [610.12]

selective listing in combination (SLIC) index. An automatic index in which the entries are combinations of terms taken from a set of preselected keywords. [610.2]

selective trace. A variable trace that involves only selected variables. *See also:* **execution trace; retrospective trace; subroutine trace; symbolic trace; variable trace.** [610.12]

self-checking code. *See:* **error detecting code.** [610.1]

self-complementing code. A binary code in which the complement of each decimal digit represented equals the complement of its binary representation. *See also:* **excess-three code.** [610.1]

self-contained. Pertaining to a database management system having a programming language that contains all of the necessary facilities for the control and processing of a database. [610.5]

self-descriptiveness. The degree to which a system or component contains enough information to explain its objectives and properties. *See also:* **maintainability; testability; usability.** [610.12]

self-documented. Pertaining to source code that contains comments explaining its objectives, operation, and other information useful in understanding and maintaining the code. [610.12]

self-relative address. An address that must be added to the address of the instruction in which it appears to obtain the address of the storage location to be accessed. *See also:* **base address; indexed address; offset; relative address.** [610.12]

semantic error. An error resulting from a misunderstanding of the relationship of symbols or groups of symbols to their meanings in a given language. *Contrast with:* **syntactic error.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

semantics. The relationships of symbols or groups of symbols to their meanings in a given language. *Contrast with:* **syntax.** [610.12]

semaphore. A shared variable used to synchronize concurrent processes by indicating whether an action has been completed or an event has occurred. *See also:* **flag; indicator.** [610.12]

semi-manual hyphenation. In text formatting, hyphenation in which most line-ending and word break decisions are made automatically, the user being asked to assist only when a determination cannot be made automatically. *See also:* **automatic hyphenation; hot-zone hyphenation; manual hyphenation.** [610.2]

semi-Markov model. A Markov chain model in which the length of time spent in each state is randomly distributed. [610.3]

semi-Markov process. A Markov process in which the duration of each event is randomly distributed. [610.3]

senary. (1) Pertaining to a selection in which there are six possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of 6. [610.1]

separate chaining. Hashing in which collision resolution is handled by building a linked list, called a collision chain, for each position in the hash table to hold the items whose hash values correspond to that position in the hash table. *Syn:* **direct chaining; external chaining.** *Contrast with:* **open-address hashing.** [610.5]

separating character. *See:* **information separator.** [610.5]

separation sort. *See:* **distribution sort.** [610.5]

separator. *See:* **delimiter.** [610.5]

septenary. (1) Pertaining to a selection in which there are seven possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of 7. [610.1]

septendecimal. (1) Pertaining to a selection in which there are 17 possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of 17. [610.1]

septet. A group of seven adjacent digits operated upon as a unit. *Syn:* **seven-bit byte.** [610.1, 610.5]

sequence. (1) To place items in a linear arrangement in accordance with the order of the natural numbers. *Note:* Methods or procedures may be specified for other natural linear orders by mapping onto the natural numbers. For example, the sequence may be alphabetic or chronological. *See also:* **collating sequence (2); sort (1).** [610.5]
(2) The order (2) in which items are arranged. *See also:* collating sequence (1); random number sequence; recursively defined sequence. [610.5]
(3) A set of items that have been sequenced. [610.5]
(4) * *See:* **order (1).** [610.5]
(5) *See:* **collating sequence (2).** [610.5]
* Deprecated.

sequence by merging. *See:* **sort by merging.** [610.5]

sequence check. A check that verifies that a set of items are in a certain sequence. [610.5]

sequence field. *See:* **key.** [610.5]

sequencer. A mechanical device or computer program that sequences the items in a set. *See also:* **sorter.** [610.5]

sequencing key. *See:* **sort key.** [610.5]

sequential. Pertaining to the occurrence of two or more events or activities in such a manner that one must finish before the next begins. *Syn:* **serial (2).** *See also:* **consecutive.** [610.12]

sequential access. Pertaining to the process of storing and retrieving data using the sequential access mode. *Syn:* **physical sequential access; serial access.** *Contrast with:* **direct access; random access.** *See also:* **indexed access; indexed sequential access.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

sequential access method (SAM). A technique for accessing data using sequential access mode. That is, to process a given data record, all data records previous to it must be accessed. *See also*: **basic sequential access method; queued sequential access method.** [610.5]

sequential access mode. An access mode in which data records are stored and retrieved in such a way that each successive access defines the next record to be retrieved. *Contrast with*: **direct access mode; indexed sequential access mode.** [610.5]

sequential cohesion. A type of cohesion in which the output of one task performed by a software module serves as input to another task performed by the module. *Contrast with:* **coincidental cohesion; communicational cohesion; functional cohesion; logical cohesion; procedural cohesion; temporal cohesion.** [610.12]

sequential construct. *See:* **serial construct.** [610.12]

sequential data set (SDS). *See*: **sequential file.** [610.5]

sequential file. A file that must be accessed using sequential access; for example, a data file on a magnetic tape. *Syn*: **serial file; sequential data set.** *Contrast with*: **direct data set; indexed file; partitioned data set.** [610.5]

sequential precedential database. *See*: **hierarchical database.** [610.5]

sequential search. A search in which the items in a set are examined in order, starting from the first item in the set, until the search is successful or the end of the set is encountered. *Syn*: **linear search.** [610.5]

serial. (1) Pertaining to the sequential transfer, occurrence, or processing of the individual parts of a whole, such as the bits of a character, using the same facilities for successive parts. *Contrast with:* **parallel (1).** [610.12]
(2) *See:* **sequential.** [610.12]

serial access. *See*: **sequential access.** [610.5]

serial addition. Addition that is performed by adding the corresponding digits of the operands, one digit place at a time. *Contrast with:* **parallel addition.** [610.1]

serial construct. A program construct consisting of a sequence of steps not involving a decision or loop. *Syn:* **sequential construct.** [610.12]

serial file. *See*: **sequential file.** [610.5]

serially-correlated variable. *See:* **lag variable.** [610.3]

set. (1) In a CODASYL model or network model, a named collection of records. *Syn*: **CODASYL set.** [610.5]
(2) In database design, a finite or infinite number of objects of any kind, of entities, or of concepts, that have a given property or properties in common. [610.5]

set difference. *See*: **difference.** [610.5]

set-up time. The period of time during which a system or component is being prepared for a specific operation. *See also:* **busy time; down time; idle time; up time.** [610.12]

seven-bit byte. *See*: **septet.** [610.5]

severity. *See:* **criticality.** [610.12]

sexadecimal. *See:* **hexadecimal.** [610.1]

sexagenary. (1) Pertaining to a selection in which there are 60 possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of 60. *Syn:* **sexagesimal.** [610.1]

sexagesimal. *See:* **sexagenary.** [610.1]

sextet. A group of six adjacent digits operated upon as a unit. *Syn*: **six-bit byte.**[610.1, 610.5]

shared lock. A lock that allows several processes concurrent access to data. *Note*: at most, only one of the processes is allowed to modify the data and the other processes may only read the data. *Contrast with*: **exclusive lock.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052922 9

shared-logic word processing. Word processing performed on a system composed of multiple work stations that share the logic and storage sections of a single central processor. *Contrast with:* **clustered word processing; dedicated word processing; shared-resource word processing; stand-alone word processing.** [610.2]

shared-resource word processing. Word processing performed on a system composed of multiple work stations, each with its own processor but sharing certain resources such as printers and disk drives. *Contrast with:* **clustered word processing; dedicated word processing; shared-logic word processing; stand-alone word processing.** [610.2]

sharp. Pertaining to elements in an image that are well defined and readily discernable. *Contrast with:* **blurred.** [610.4]

sharpening. Any image enhancement technique in which the effect of blurring in the original image is reduced. *Syn:* **deblurring.** *See also:* **unsharp masking.** [610.4]

Sheffer stroke. *See:* **NAND.** [610.1]

Sheffer stroke function. *See:* **NAND operation.** [610.1]

shell. A computer program or routine that provides an interface between the user and a computer system or program. [610.12]

Shell's method. *See*: **diminishing increment sort.** [610.5]

Shell sort. *See*: **diminishing increment sort.** [610.5]

shift. A displacement of an ordered set of characters one or more places to the left or right. If the characters are the digits of a numeral, a shift may be equivalent to multiplying by a power of the base. *See also:* **arithmetic shift; logical shift.** [610.1]

shift character. A control character that determines the alphabetic or numeric shift of character codes in a message. [610.5]

shift-in character (SI). A code extension character, used to terminate a sequence that has been introduced by the shift-out character, that makes effective the graphic characters of the original character set. *Contrast with*: **shift-out character.** [610.5]

shift-out character (SO). A code extension character that substitutes, for the graphic characters of the original character set, an alternative set of graphic characters upon which agreement has been reached or that has been designated using code extension procedures. *Contrast with*: **shift-in character.** [610.5]

short answer interaction. An instruction method employed by some computer-assisted instruction systems, in which the student is asked to provide a word or phrase in response to a question. *See also:* **question-and-answer interaction.** [610.2]

SI. Acronym for the **shift-in character.** [610.5]

sibling node. Relative to a node in a tree, a second node that has the same immediate predecessor or parent node. *Syn*: **brother; sister.** [610.5]

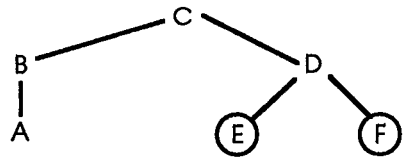


Fig 64
Nodes E and F are Sibling Nodes

sideways sum. A sum obtained by adding the digits of a numeral without regard to position or significance. *See also:* **check sum.** [610.1]

sifting sort. *See*: **bubble sort.** [610.5]

signal distance. *See:* **Hamming distance.** [610.1]

sign bit. A binary digit used to indicate the algebraic sign of a number. [610.1]

sign character. A character within a picture specification that represents the sign of a

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

data item. *Note*: S, +, and - are commonly used as sign characters. [610.5]

sign digit. A character used to represent the algebraic sign of a number. [610.1]

signed. Pertaining to a representation of a number with which an algebraic sign is associated. [610.5]

signed binary arithmetic. *See:* **sign-magnitude arithmetic.** [610.1]

significance. *See:* **weight.** [610.1]

significand. The component of a floating-point number that consists of an explicit or implicit leading digit to the left of its implied radix point and a fraction field to the right. *Syn:* **fixed-point part; mantissa.** *Contrast with:* **exponent (2).** [610.1]

significant code. A code that identifies a particular item and also yields further information about the properties or classification of the item. *Contrast with*: **non-significant code.** [610.5]

significant digit. A digit that contributes to the accuracy or precision of a numeral. *See also:* **least significant digit; most significant digit.** [610.1, 610.5]

significant-digit arithmetic. A method of making calculations using a modified form of floating-point representation in which the number of significant digits in the result is determined by the number of significant digits in the operands, the operations performed, and the degree of precision available. [610.1]

significant figure.* *See:* **significant digit.**
* Deprecated. [610.1, 610.5]

sign-magnitude arithmetic. Computer arithmetic using numerals expressed in sign-magnitude notation. *Syn:* **signed binary arithmetic.** [610.1]

sign-magnitude notation. A numeration system in which the left-most bit is interpreted as the sign bit and the remaining bits represent the magnitude. *Contrast with:* **twos-complement notation.** [610.1]

sign position. The position at which the sign of a number is located. [610.1]

simple arc. An arc that does not cross itself. [610.4]

simple buffering. A buffering technique in which a buffer is allocated to a computer program for the duration of the program's execution. *Contrast with:* **dynamic buffering.** [610.12]

simple path. A path in which all vertices except the first and last in the sequence are distinct. [610.5]

simplicity. The degree to which a system or component has a design and implementation that is straightforward and easy to understand. *Contrast with:* **complexity.** [610.12]

simuland. The system being simulated by a simulation. [610.3]

simulate. To represent a system by a model that behaves or operates like the system. *See also:* **emulate.** [610.3]

simulated time. Time as represented within a simulation. *Syn:* **virtual time.** *See also:* **fast time; real time; slow time.** [610.3]

simulation. (1) A model that behaves or operates like a given system when provided a set of controlled inputs. *Syn:* **simulation model.** *See also:* **emulation.** [610.3, 610.12]
(2) The process of developing or using a model as in (1). [610.3, 610.12]
(3) An instruction method employed by some computer-assisted instruction systems, in which a situation is simulated and the student must respond appropriately. *Contrast with:* **instructional game; question-and-answer interaction.** [610.2]

simulation clock. A counter used to accumulate simulated time. [610.3]

simulation game. A simulation in which the participants seek to achieve some agreed-upon objective within an established set of rules. For example, a management game, a war game. *Note:* The objective may not be to

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

compete, but to evaluate the participants, increase their knowledge concerning the simulated scenario, or achieve other goals. *Syn:* **gaming simulation.** [610.3]

simulation language. A programming language used to implement simulations.[610.3]

simulation model. *See:* **simulation.** [610.3]

simulator. A device, computer program, or system that performs simulation. *See also:* **emulator.** [610.3, 610.12a]

simultaneous. Pertaining to the occurrence of two or more events at the same instant of time. *Contrast with:* **concurrent.** [610.12]

simultaneous access. *See*: **immediate access.** [610.5]

simultaneous recursion. A situation in which two software modules call each other.[610.12]

single-address instruction. *See:* **one-address instruction.** [610.12]

single-font character recognition. Character recognition of one character font. *Contrast with:* **omni-font character recognition.**[610.2]

single-level encoding. A microprogramming technique in which different microoperations are encoded as different values in the same field of a microinstruction. *Contrast with:* **two-level encoding.** [610.12]

single-operand instruction. *See:* **one-address instruction.** [610.12]

single precision. Pertaining to the use of a single computer word to represent a number. *Note:* Single precision is implied in number representation and in computer arithmetic unless multiple precision is specified. *Contrast with*: **double precision; multiple precision; triple precision.** [610.1, 610.5]

single-step execution. *See:* **single-step operation.** [610.12]

single-step operation. A debugging technique in which a single computer instruction, or part of an instruction, is executed in response to an external signal. *Syn:* **single-step execution; step-by-step operation.**[610.12]

singly linked list. *See*: **linked list.** [610.5]

sister. *See*: **sibling node.** [610.5]

six-bit byte. *See*: **sextet.** [610.5]

sizing. The process of estimating the amount of computer storage or the number of source lines required for a software system or component. *Contrast with:* **timing.** [610.12]

slave state. *See:* **problem state.** [610.12]

SLIC. Acronym for **selective listing in combination.** *See:* **selective listing in combination index.** [610.2]

slow time. (1) Simulated time with the property that a given period of actual time represents less than that period of time in the system being modeled; for example, in a simulation of the internal workings of a computer, running the simulation for one second may result in the model advancing time by only a microsecond; that is, simulated time advances slower than actual time. [610.3] (2) The duration of activities within a simulation in which simulated time advances slower than actual time. *Contrast with:* **fast time; real time.** [610.3]

smooth. To apply procedures that decrease or eliminate rapid fluctuations in data. [610.1]

smoothing. Any image enhancement technique in which the effect of noise in the original image is reduced. *Syn:* **noise cleaning; noise suppression.** [610.4]

snapshot. A copy of all or portions of the data contained in storage or in a database at a particular point in time. *Note*: Considered a "picture" of the data. [610.5]

snapshot dump. A dynamic dump of the contents of one or more specified storage areas. *See also:* **change dump; dynamic dump; memory dump; postmortem dump; selective dump; static dump.** [610.12]

SO. Acronym for the **shift-out character.**[610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

soft copy. A copy of computer output in a form other than a printed page. For example, data displayed on a video display device. *Contrast with:* **hard copy.** [610.2]

soft error.* *See:* **transient error.** [610.12]
* Deprecated.

soft failure. A failure that permits continued operation of a system with partial operational capability. *Contrast with:* **hard failure.** [610.12]

soft hyphen. *See:* **discretionary hyphen.** [610.2]

software. Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system. *See also:* **application software; support software; system software.** *Contrast with:* **hardware.** [610.12]

software characteristic. An inherent, possibly accidental, trait, quality, or property of software (for example, functionality, performance, attributes, design constraints, number of states, lines or branches). [610.12]

software configuration management. *See:* **configuration management.** [610.12]

software design description (SDD). (1) A representation of software created to facilitate analysis, planning, implementation, and decision making. The software design description is used as a medium for communicating software design information, and may be thought of as a blueprint or model of the system. [610.12]
(2) A representation of a software system created to facilitate analysis, planning, implementation, and decision making. A blueprint or model of the software system. The SDD is used as the primary medium for communicating software design information. [610.12]

software development cycle. The period of time that begins with the decision to develop a software product and ends when the software is delivered. This cycle typically includes a requirements phase, design phase, implementation phase, test phase, and sometimes, installation and checkout phase. *Contrast with:* **software life cycle.** [610.12]
Notes: (1) The phases listed above may overlap or be performed iteratively, depending upon the software development approach used. [610.12]
(2) This term is sometimes used to mean a longer period of time, either the period that ends when the software is no longer being enhanced by the developer, or the entire software life cycle. [610.12]

software development file (SDF). A collection of material pertinent to the development of a given software unit or set of related units. Contents typically include the requirements, design, technical reports, code listings, test plans, test results, problem reports, schedules, and notes for the units. *Syn:* **software development folder; software development notebook; unit development folder.** [610.12]

software development folder. *See:* **software development file.** [610.12]

software development library. A software library containing computer readable and human readable information relevant to a software development effort. *Syn:* **project library; program support library.** *Contrast with:* **master library; production library; software repository; system library.** [610.12]

software development notebook. *See:* **software development file.** [610.12]

software development plan (SDP). A project plan for a software development project. [610.12]

software development process. The process by which user needs are translated into a software product. The process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code, and sometimes, installing and checking out the software for operational use. *Note:* These activities may overlap or be performed iteratively. *See also:* **incremental development; rapid prototyping; spiral model; waterfall model.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052926 6

software diversity. A software development technique in which two or more functionally identical variants of a program are developed from the same specification by different programmers or programming teams with the intent of providing error detection, increased reliability, additional documentation, or reduced probability that programming or compiler errors will influence the end results. *See also:* **diversity.** [610.12]

software engineering. (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. [610.12]
(2) The study of approaches as in (1). [610.12]

software engineering environment. The hardware, software, and firmware used to perform a software engineering effort. Typical elements include computer equipment, compilers, assemblers, operating systems, debuggers, simulators, emulators, test tools, documentation tools, and database management systems. [610.12]

software feature. (1) A distinguishing characteristic of a software item (for example, performance, portability, or functionality). [610.12]
(2) A software characteristic specified or implied by requirements documentation (for example, functionality, performance, attributes, or design constraints). [610.12]

software item. Source code, object code, job control code, control data, or a collection of these items. [610.12]

software library. A controlled collection of software and related documentation designed to aid in software development, use, or maintenance. Types include master library, production library, software development library, software repository, system library. *Syn:* **program library.** [610.12]

software life cycle. The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The software life cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and, sometimes, retirement phase. *Note:* These phases may overlap or be performed iteratively. *Contrast with:* **software development cycle.** [610.12]

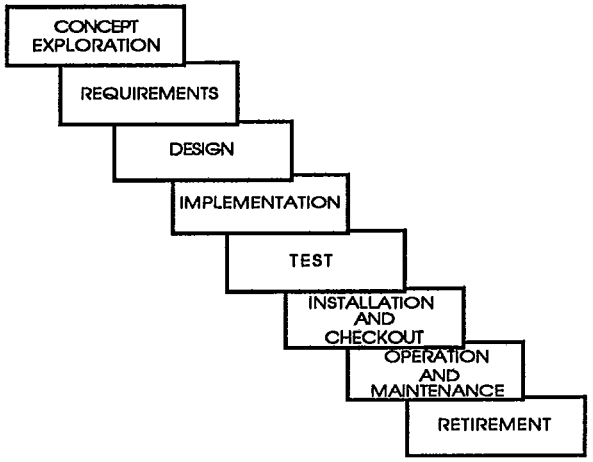


Fig 65
Sample Software Life Cycle

software maintenance. *See:* **maintenance (1).** [610.12]

software model. A symbolic model whose properties are expressed in software; for example, a computer program that models the effects of climate on the world economy. *Contrast with:* **graphical model; mathematical model; narrative model; tabular model.** [610.3]

software monitor. A software tool that executes concurrently with another program and provides detailed information about the execution of the other program. *See also:* **hardware monitor; monitor.** [610.12]

software product. (1) The complete set of computer programs, procedures, and possibly associated documentation and data designated for delivery to a user. [610.12]
(2) Any of the individual items in (1). [610.12]

software quality assurance. *See:* **quality assurance.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

software quality metric. *See:* **quality metric.** [610.12]

software repository. A software library providing permanent, archival storage for software and related documentation. *Contrast with:* **master library; production library; software development library; system library.** [610.12]

software requirements review (SRR). (1) A review of the requirements specified for one or more software configuration items to evaluate their responsiveness to and interpretation of the system requirements and to determine whether they form a satisfactory basis for proceeding into preliminary design of the configuration items. *See also:* **system requirements review.** *Note:* This review is called software specification review by the U.S. Department of Defense. [610.12]
(2) A review as in (1) for any software component. [610.12]

software requirements specification (SRS). Documentation of the essential requirements (functions, performance, design constraints, and attributes) of the software and its external interfaces. [610.12]

software specification review (SSR). *See:* **software requirements review.** [610.12]

software test incident. Any event occurring during the execution of a software test that requires investigation. [610.12]

software tool. A computer program used in the development, testing, analysis, or maintenance of a program or its documentation. Examples include comparator, cross-reference generator, decompiler, driver, editor, flowcharter, monitor, test case generator, timing analyzer. [610.12]

son. *See*: **child node.** [610.5]

son file. A file that contains data that have been updated from those in another file, called the father file. *See also*: **grandfather file.** [610.5]

sort. (1) To arrange items according to a specified order of their sort keys. For example, to arrange the records of a personnel file into alphabetical sequence using the sort key "Employee-name." *See also*: **distribution sort; exchange sort; external sort; insertion sort; internal sort; merge sort; radix sort; selection sort.** [610.5]
(2) To segregate items into subsets according to specified criteria. [610.5]
(3) A process that achieves the arrangement or segregation described in (1) or (2). [610.5]

sort by merging. To sort the items of a set by splitting the set into subsets, sorting the subsets, and merging the subsets. *Syn*: **sequence by merging.** *See also*: **order by merging.** [610.5]

sorter. A mechanical device that deposits punched cards in pockets based on the hole patterns in the cards. [610.5]

sorting item. (1) That item of a set that is actively being exchanged or manipulated with other elements during the sorting process. *See also*: **sort selection.** [610.5]
(2) Any element of a set that has a probability of being selected by a sort selection. [610.5]

sorting rewind time. In a tape merge sort, the length of time needed to rewind a tape to its original position. [610.5]

sorting-sequencing key. *See*: **sort key.** [610.5]

sorting string. A string of characters used as a sort key. [610.5]

sort key. A key field whose value is used to determine the position of items within a sorted set. *See also*: **sort (2); sorting string.** *Syn*: **sequencing key; sorting-sequencing key.** [610.5]

sort order. * *See*: **order (2).** [610.5]
* Deprecated.

sort pass. (1) In a sorting algorithm, a single processing of all the items of a set. [610.5]
(2) A phase of a merge sort that reads a subset of unsorted data items, orders them, and places the ordered subset on a data medium. This process is repeated until all input data is placed in some subset. The merge phase is then begun to merge the subsets into one ordered set. [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

sort selection. (1) The choice of a particular sorting algorithm. [610.5]
(2) The process of choosing an item to be exchanged with another item as part of a selection sorting process. *See*: **sorting item.** [610.5]

source address. The address of a device or storage location from which data is to be transferred. *Contrast with:* **destination address.** [610.12]

source code. Computer instructions and data definitions expressed in a form suitable for input to an assembler, compiler, or other translator. *Note:* A source program is made up of source code. *Contrast with:* **object code.** [610.12]

source code generator. *See:* **code generator (2).** [610.12]

source document. A document containing information that is to be input to a computer. For example, an original invoice, a library charge-out card, or a machine-readable document. [610.2]

source language. The language in which the input to a machine-aided translation process is represented. For example, the language used to write a computer program. *Contrast with:* **target language.** [610.2, 610.12a]

source program. A computer program that must be compiled, assembled, or otherwise translated in order to be executed by a computer. *Contrast with:* **object program.** [610.12]

source segment. In a hierarchical database, a segment that contains the data used to construct a secondary index. [610.5]

SP. Abbreviation for the **space character.** [610.5]

SPA. Acronym for **scratchpad area.** [610.5]

space. (1) A site intended for the storage of data such as a location in a storage medium. [610.5]
(2) A basic unit of area such as the size of a single character. [610.5]
(3) One or more space characters. [610.5]

space character (SP). A graphic character that is usually represented by a blank site in a series of graphics. The space character, though not a control character, has the function equivalent to that of a format effector that causes the print or display position to move one position forward without producing the printing or display of any graphic. Similarly, the space character may have a function equivalent to that of an information separator. *See also*: **null character; space.** [610.5]

spanned record. A record that is partially contained in more than one block; that is, it spans a block boundary. *See also*: **blocked record; unblocked record.** [610.5]

special addition.* *See:* **double-precision addition.** [610.1]
*Deprecated.

special character. (1) A character that is not in the alphabet, but that is used for punctuation or another special purpose. For example, blank, comma, period, or asterisk. [610.5]
(2) A graphic character in a character set that is not a letter, not a digit, and not a space character. [610.5]
(3) In COBOL, a character that is neither numeric nor alphabetic. [610.5]

specific address. *See:* **absolute address.** [610.12]

specification. A document that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a system or component, and, often, the procedures for determining whether these provisions have been satisfied. *See also:* **formal specification; product specification; requirements specification.** [610.12]

specification change notice (SCN). A document used in configuration management to propose, transmit, and record changes to a specification. *See also:* **configuration control; engineering change; notice of revision.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

specification language. A language, often a machine-processible combination of natural and formal language, used to express the requirements, design, behavior, or other characteristics of a system or component. For example, a design language or requirements specification language. *Contrast with:* **programming language; query language.** [610.12]

specification tree. A diagram that depicts all of the specifications for a given system and shows their relationships to one another. *See also:* **documentation tree.** [610.12]

specific code. *See:* **absolute code.** [610.12]

spiral model. A model of the software development process in which the constituent activities, typically requirements analysis, preliminary and detailed design, coding, integration, and testing, are performed iteratively until the software is complete. *Contrast with:* **waterfall model.** *See also:* **incremental development; rapid prototyping.** [610.12]

spool. To read input data, or write output data, to auxiliary or main storage for later processing or output, in order to permit input/output devices to operate concurrently with job execution. Derived from the acronym SPOOL for Simultaneous Peripheral Output On Line. [610.12]

spooler. A program that initiates and controls spooling. [610.12]

spread sheet. *See:* **electronic spread sheet.** [610.2]

SRR. (1) Acronym for **software requirements review.** [610.12]
(2) (DoD) Acronym for system requirements review. [610.12]

SRS. Acronym for **software requirements specification.** [610.12]

SSR. Acronym for **software specification review.** *See:* **software requirements review.** [610.12]

stabilized-variable model. A model in which some of the variables are held constant and the others are allowed to vary; for example, a model of a controlled climate in which humidity is held constant and temperature is allowed to vary. [610.3]

stack. (1) A list in which items are appended to and retrieved from the same end of the list, known as the top. That is, the next item to be retrieved is the item that has been in the list for the shortest time. *Syn:* **push-down list; push-down stack; push-down storage; storage stack.** *Contrast with:* **queue.** [610.5]
(2) A line formed by items waiting for service in a system in which the next item to exit the line is the item that has been in the line for the shortest time. [610.5]
(3) To arrange in, or to form a stack as in (1). [610.5]

stack indicator. *See:* **stack pointer.** [610.5]

stack pointer. A data item that specifies the address of the data item most recently stored in a stack. *Syn:* **stack indicator.** [610.5]

stand-alone. Pertaining to hardware or software that is capable of performing its function without being connected to other components; for example, a stand-alone word processing system. [610.12]

stand-alone data dictionary. *See:* **passive data dictionary.** [610.5]

stand-alone word processing. Word processing performed on a system that does not depend on the resources of other equipment to perform word processing activities. *Contrast with:* **dedicated word processing; shared-logic word processing; shared-resource word processing.** [610.2]

standard binary. *See:* **binary.** [610.1]

standard form. *See:* **normalized form.** [610.5]

standardize. *See:* **normalize.** [610.1, 610.5]

standards. Mandatory requirements employed and enforced to prescribe a disciplined uniform approach to software development, that is, mandatory conventions and practices are in fact standards. *See also:* **practices; standards.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

standby redundancy. In fault tolerance, the use of redundant elements that are left inoperative until a failure occurs in a primary element. *Contrast with:* **active redundancy.** [610.12]

standby time. *See:* **idle time.** [610.12]

standing-on-nines carry. A carry process in which a carry digit transferred to a given digit place is further transferred to the next higher digit place if the current sum in the given digit place is nine. [610.1]

starting address. The address of the first instruction of a computer program in main storage. *Note:* This address may or may not be the same as the program's origin, depending upon whether there are data preceding the first instruction. *Contrast with:* **origin.** *See also:* **assembled origin; loaded origin.** [610.12]

start signal. A signal at the beginning of a start-stop character that prepares the receiving device for the reception of the code elements. *Note*: A start signal is limited to one signal element generally having the duration of unit interval. [610.5]

start-stop character. A character including one start signal at the beginning and one or two stop signals at the end. [610.5]

state. (1) A condition or mode of existence that a system, component, or simulation may be in; for example, the pre-flight state of an aircraft navigation program or the input state of given channel. [610.3, 610.12]
(2) The values assumed at a given instant by the variables that define the characteristics of a system, component, or simulation. *Syn:* **system state.** *See also:* **final state; initial state; steady state.** [610.3, 610.12]

state data. Data that defines an internal state of the test unit and is used to establish that state or compare with existing states. [610.12]

state diagram. A diagram that depicts the states that a system or component can assume, and shows the events or circumstances that cause or result from a change from one state to another. [610.12]

state machine. A model of a system in which all values are discrete, as in a digital computer. [610.3]

statement. In a programming language, a meaningful expression that defines data, specifies program actions, or directs the assembler or compiler. *See also:* **assignment statement; control statement; declaration.** [610.12]

statement testing. Testing designed to execute each statement of a computer program. *Contrast with:* **branch testing; path testing.** [610.12]

state transition. A change from one state to another in a system, component, or simulation. [610.3]

state transition diagram. *See:* **state diagram.** [610.12]

state variable. A variable that defines one of the characteristics of a system, component, or simulation. The values of all such variables define the state of the system, component, or simulation. [610.3]

static. Pertaining to an event or process that occurs without computer program execution; for example, static analysis, static binding. *Contrast with:* **dynamic.** [610.12]

static analysis. The process of evaluating a system or component based on its form, structure, content, or documentation. *Contrast with:* **dynamic analysis.** *See also:* **inspection; walk-through.** [610.12]

static binding. Binding performed prior to the execution of a computer program and not subject to change during program execution. *Contrast with:* **dynamic binding.** [610.12]

static breakpoint. A breakpoint that can be set at compile time, such as entry into a given routine. *Contrast with:* **dynamic breakpoint.** *See also:* **code breakpoint; data breakpoint; epilog breakpoint; programmable breakpoint; prolog breakpoint.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

static dump. A dump that is produced before or after the execution of a computer program. *Contrast with:* **dynamic dump.** *See also:* **change dump; memory dump; postmortem dump; selective dump; snapshot dump.** [610.12]

static error. An error that is independent of the time-varying nature of an input. *Contrast with:* **dynamic error.** [610.12]

static model. A model of a system in which there is no change; for example, a scale model of a bridge, studied for its appearance rather than for its performance under varying loads. *Contrast with:* **dynamic model.** [610.3]

statistical pattern recognition. An approach to pattern recognition that uses probability and statistical methods to assign patterns to pattern classes. [610.4]

status code. A code used to indicate the results of a computer program operation. For example, a code indicating a carry, an overflow, or a parity error. *Syn:* **condition code.** [610.12]

steady state. A situation in which a model, process, or device exhibits stable behavior independent of time. *Syn:* **equilibrium.** [610.3]

step-by-step operation. *See:* **single-step operation.** [610.12]

stepwise refinement. A software development technique in which data and processing steps are defined broadly at first and then further defined with increasing detail. *See also:* **data structure-centered design; input-process-output; modular decomposition; object-oriented design; rapid prototyping; structured design; transaction analysis; transform analysis.** [610.12]

stimulate. To provide input to a system in order to observe or evaluate the system's response. [610.3]

stochastic. (1) Pertaining to a process, model, or variable whose outcome, result, or value depends on chance. *Contrast with:* **deterministic.** [610.3]

(2) Pertaining to variables that are probabilistic in nature. [610.1]

stochastic model. A model in which the results are determined by using one or more random variables to represent uncertainty about a process or in which a given input will produce an output according to some statistical distribution; for example, a model that estimates the total dollars spent at each of the checkout stations in a supermarket, based on probable number of customers and probable purchase amount of each customer. *Syn:* **probabilistic model.** *See also:* **Markov-chain model.** *Contrast with:* **deterministic model.** [610.3]

stop. To terminate the execution of a computer program. *Syn:* **halt (1).** *Contrast with:* **pause.** [610.12]

stop character. A word processing control character that interrupts the sequence of output processing to provide the ability to make changes in the text formatting parameters, the text itself, the character font on the output device, or other items. *Syn:* **stop code.** [610.2]

stop code. *See:* **stop character.** [610.2]

stop list. In automatic indexing, a list of terms, words, or roots of words that are considered insignificant for purposes of information retrieval, and are excluded from being keywords in an index. *Contrast with:* **go list.** *Syn:* **stop word list.** [610.2]

stop signal. A signal at the end of a start-stop character that prepares the receiving device for the reception of a subsequent character. *Note*: A stop signal is usually limited to one signal element having any duration equal to or greater than a specified minimum value. [610.5]

stopword list. *See:* **stop list.** [610.2]

storage access. *See*: **access.** [610.5]

storage. In a computer, one or more bytes that are used to store data. [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** ■ 4805702 0052932 1 ■

storage allocation. An element of computer resource allocation, consisting of assigning storage areas to specific jobs and performing related procedures, such as transfer of data between main and auxiliary storage, to support the assignments made. *See also:* **buffer; contiguous allocation; cyclic search; memory compaction; overlay; paging; virtual storage.** [610.12]

storage breakpoint. *See:* **data breakpoint.** [610.12]

storage capacity. The maximum number of items that can be held in a given storage device; usually measured in words or bytes. *See also:* **channel capacity; memory capacity.** [610.12]

storage efficiency. The degree to which a system or component performs its designated functions with minimum consumption of available storage. *See also:* **execution efficiency.** [610.12]

storage location. An area in a storage device that can be explicitly and uniquely specified by means of an address. [610.5]

storage schema. In a CODASYL database, statements expressed in data storage definition language that describe storage areas, stored records, and any associated indices and access paths supporting the records and sets defined by a given schema. *See also*: **CODASYL database.** [610.5]

storage stack. *See*: **stack.** [610.5]

storage structure. (1) The manner in which data structures are represented in storage. [610.5]

(2) The configuration of a database resident on computer storage devices after mapping the data elements of the logical structure of the database onto their respective physical counterparts. *Note*: The relationships and associations that provide the physical means for accessing the information stored in the database are preserved. [610.5]

store. (1) To place or retain data in a storage device. [610.5a, 610.12]

(2) To copy computer instructions or data from a register to internal storage or from internal storage to external storage. *Contrast with:* **load (2); retrieve.** *See also:* **fetch; move.** [610.5a, 610.12]

stored paragraph. *See:* **boilerplate text.** [610.2]

stored record. *See*: **internal record.** [610.5]

straight binary. *See:* **binary.** [610.1]

straight insertion sort. *See*: **insertion sort.** [610.5]

straight-line code. A sequence of computer instructions in which there are no loops. [610.12]

straight-line coding. A programming technique in which loops are avoided by stating explicitly and in full all of the instructions that would be involved in the execution of each loop. *See also:* **unwind.** [610.12]

straight line sort. *See*: **linear sort.** [610.5]

straight radix sort. A radix sort in which items are sorted repeatedly on successive digits within the numeric representation of the sort key, starting with the least significant digit. [610.5]

straight selection sort. *See*: **selection sort.** [610.5]

straight two-way merge sort. A variation of the natural two-way merge sort in which the set to be sorted is repeatedly divided into two ordered subsets of length 2 to the power of k, where k is the number of passes made so far. *Contrast with*: **natural two-way merge sort.** [610.5]

stratified language. A language that cannot be used as its own metalanguage. Examples include FORTRAN, COBOL. *Contrast with:* **unstratified language.** [610.12]

stress testing. Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements. *See also:* **boundary value.** [610.12]

string. (1) A sequence of bits, characters, or other entities; for example, the bit string

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

0101010 or the character string XYZ. *See also:* **bit string; character string.** [610.5]
(2) Pertaining to data that contains a sequence as in (1). *Contrast with:* **arithmetic.** [610.5]

stroke. In character recognition, a straight line or arc used as a segment of a graphic character. [610.2]

stroke centerline. In character recognition, a line midway between two stroke edges. [610.2]

stroke edge. In character recognition, the line of discontinuity between a side of a stroke and the background, obtained by averaging, over the length of the stroke, the irregularities resulting from the printing and detecting process. [610.2]

stroke width. In character recognition, the distance between two stroke edges, measured perpendicular to the stroke centerline. [610.2]

strong typing. A feature of some programming languages that requires the type of each data item to be declared, precludes the application of operators to inappropriate data types, and prevents the interaction of data items of incompatible types. [610.12]

structural model. A model of the physical or logical structure of a system; for example, a model that represents a computer network as a set of boxes connected by communication lines. *Contrast with:* **process model.** [610.3]

structural pattern recognition. An approach to pattern recognition in which patterns are represented in terms of primitives and relationships among those primitives in order to describe and classify pattern structure. *See also:* **syntactic pattern recognition.** [610.4]

structural testing. Testing that takes into account the internal mechanism of a system or component. Types include branch testing, path testing, statement testing. *Syn:* **glass-box testing; white-box testing.** *Contrast with:* **functional testing (1).** [610.12]

structure chart. A diagram that identifies modules, activities, or other entities in a system or computer program and shows how larger or more general entities break down into smaller, more specific entities. *Note:* The result is not necessarily the same as that shown in a call graph. *Syn:* **hierarchy chart; program structure chart.** *Contrast with:* **call graph.** [610.12]

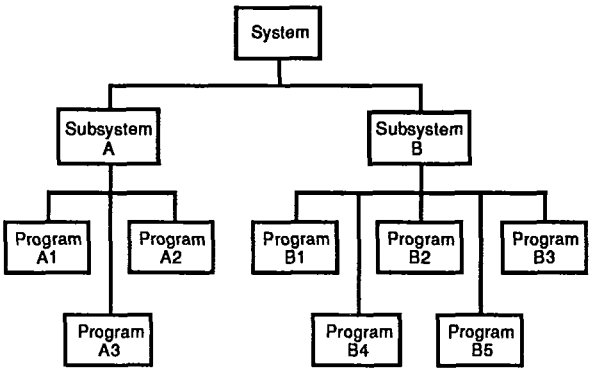


Fig 66
Structure Chart

structure clash. In software design, a situation in which a module must deal with two or more data sets that have incompatible data structures. *See also:* **data structure-centered design; order clash.** [610.12]

structured design. (1) Any disciplined approach to software design that adheres to specified rules based on principles such as modularity, top-down design, and stepwise refinement of data, system structures, and processing steps. *See also:* **data structure-centered design; input-process-output; modular decomposition; object-oriented design; rapid prototyping; stepwise refinement; transaction analysis; transform analysis.** [610.12]
(2) The result of applying the approach in (1). [610.12]

structured program. A computer program constructed of a basic set of control structures, each having one entry and one exit. The set of control structures typically includes: sequence of two or more instructions, conditional selection of one of two or more sequences of instructions, and repetition of a sequence of instructions. *See also:* **structured design.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052934 5

structured programming language. A programming language that provides the structured program constructs, namely, single-entry-single-exit sequences, branches, and loops, and facilitates the development of structured programs. *See also:* **block-structured language.** [610.12]

structured programming. Any software development technique that includes structured design and results in the development of structured programs. [610.12]

stub. (1) A skeletal or special-purpose implementation of a software module, used to develop or test a module that calls or is otherwise dependent on it. [610.12]
(2) A computer program statement substituting for the body of a software module that is or will be defined elsewhere. [610.12]

SUB. Abbreviation for the **substitute character.** [610.5]

subdatabase. A subset of the data contained in a database as used for a specific type of application or system. [610.5]

subgraph. A graph consisting of a subset of nodes from a larger graph. [610.5]

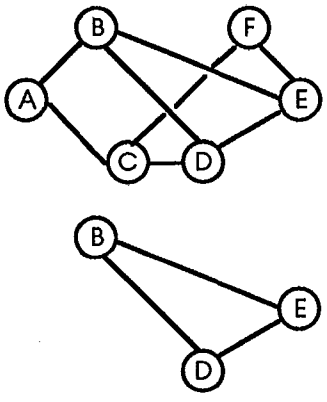


Fig 67
Graph and Subgraph

submodel. *See*: **subschema.** [610.5]

subnormal number. A non-zero floating-point number whose exponent is the precision's minimum and whose leading significant digit is zero. [610.1]

subprogram. A separately compilable, executable component of a computer program. *Note:* The terms "routine," "subprogram," and "subroutine" are defined and used differently in different programming languages; the preceding definition is advanced as a proposed standard. *See also:* **coroutine; main program; routine; subroutine.** [610.12]

subroutine. A routine that returns control to the program or subprogram that called it. *Note:* The terms "routine," "subprogram," and "subroutine" are defined and used differently in different programming languages; the preceding definition is advanced as a proposed standard. *Contrast with:* **coroutine.** *See also:* **closed subroutine; open subroutine.** [610.12]

subroutine trace. A record of all or selected subroutines or function calls performed during the execution of a computer program and, optionally, the values of parameters passed to and returned by each subroutine or function. *Syn:* **call trace.** *See also:* **execution trace; retrospective trace; subroutine trace; symbolic trace; variable trace.** [610.12]

subschema. (1) A subset of a schema that defines a view of the database that is needed by one or more application programs. *Syn*: **submodel.** *See also*: **external model.** [610.5]
(2) A description of the logical structure of a record in a database. [610.5]

subscript. A symbol that is associated with the name of a set to identify a particular subset or element of the set. [610.5]

substitute character (SUB). A control character used in the place of a character that is recognized to be invalid or in error, or that cannot be represented on a given device. [610.5]

subsystem. A secondary or subordinate system with a larger system. [610.12]

subtrahend. A number to be subtracted from another number (the minuend) to produce a result (the difference). [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

subtree. A tree whose root node is part of a larger tree. *Note*: A subtree is made up of a node and all of its hierarchical descendants. *Syn*: **branch.** [610.5]

subtype. A subset of a data type, obtained by constraining the set of possible values of the data type. *Note:* The operations applicable to the subtype are the same as those of the original data type. *See also:* **derived type.** [610.12]

suffix notation. *See:* **postfix notation.** [610.1]

sum. The result of an addition operation. [610.1]

sum check. *See:* **summation check.** [610.1]

summation check. A check in which a group of digits is summed, usually without regard to overflow, and that sum is checked against a previously computed value to verify that no digits have been changed. *Syn:* **sum check.** [610.1]

supervisor. *See:* **supervisory program.** [610.12]

supervisor state. In the operation of a computer system, a state in which the supervisory program is executing. This state usually has higher priority than, and precludes the execution of, application programs. *Syn:* **executive state; master state; privileged state.** *Contrast with:* **problem state.** [610.12]

supervisory program. A computer program, usually part of an operating system, that controls the execution of other computer programs and regulates the flow of work in a computer system. *Syn:* **control program; executive; executive program; supervisor.** *See also:* **supervisor state.** [610.12]

support. The set of activities necessary to ensure that an operational system or component fulfills its original requirements and any subsequent modifications to those requirements. For example, software or hardware maintenance, user training. *See also:* **software life cycle; system life cycle.** [610.12]

support manual. A document that provides the information necessary to service and maintain an operational system or component throughout its life cycle. Typically described are the hardware and software that make up the system or component and procedures for servicing, repairing, or reprogramming it. *Syn:* **maintenance manual.** *See also:* **diagnostic manual; installation manual; operator manual; programmer manual; user manual.** [610.12]

support software. Software that aids in the development or maintenance of other software, for example, compilers, loaders, and other utilities. *Contrast with:* **application software.** *See also:* **system software.** [610.12]

swap. (1) An exchange of the contents of two storage areas, usually an area of main storage with an area of auxiliary storage. *See also:* **roll in; roll out.** [610.12]
(2) To perform an exchange as in (1).[610.12]

switching function. A function that has only a finite number of possible values and whose independent variables each have only a finite number of possible values. [610.1]

switching variable. A variable that may take only a finite number of possible values or states. *Syn:* **logical variable.** [610.1]

syllable hyphen. *See:* **discretionary hyphen.** [610.2]

symbol. *See*: **code.** [610.5]

symbolic address. An address expressed as a name or label that must be translated to the absolute address of the device or storage location to be accessed. *Contrast with:* **absolute address.** [610.12]

symbolic execution. A software analysis technique in which program execution is simulated using symbols, such as variable names, rather than actual values for input data, and program outputs are expressed as logical or mathematical expressions involving these symbols. [610.12]

symbolic image. A digital image in which the value associated with each pixel is a symbol, rather than a gray level. [610.4]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

symbolic language. A programming language that expresses operations and addresses in symbols convenient to humans rather than in machine language. Examples are assembly language, high-order language. *Contrast with:* **machine language.** [610.12]

symbolic model. A model whose properties are expressed in symbols. Examples include graphical models, mathematical models, narrative models, software models, and tabular models. *Contrast with:* **physical model.** [610.3]

symbolic trace. A record of the source statements and branch outcomes that are encountered when a computer program is executed using symbolic, rather than actual, values for input data. *See also:* **execution trace; retrospective trace; subroutine trace; variable trace.** [610.12]

symbol rank. *See:* **digit place.** [610.1]

symbol table. A table that presents program symbols and their corresponding addresses, values, and other attributes. [610.12]

symmetric traversal. *See:* **inorder traversal.** [610.5]

synonym. In hashing, an item whose hash value is identical to that of another item. *See also:* **collision resolution.** [610.5]

syntactic error. A violation of the structural or grammatical rules defined for a language; for example, using the statement $B + C = A$ in Fortran, rather than the correct $A = B + C$. *Syn:* **syntax error.** *Contrast with:* **semantic error.** [610.12]

syntactic pattern recognition. A type of structural pattern recognition that identifies primitives and relationships in natural or artificial language patterns. [610.4]

syntax. The structural or grammatical rules that define how the symbols in a language are to be combined to form words, phrases, expressions, and other allowable constructs. *Contrast with:* **semantics.** [610.12]

syntax error. *See:* **syntactic error.** [610.12]

synthetic address. *See:* **generated address.** [610.12]

system. A collection of components organized to accomplish a specific function or set of functions. [610.3, 610.5, 610.12]

system design review (SDR). A review conducted to evaluate the manner in which the requirements for a system have been allocated to configuration items, the system engineering process that produced the allocation, the engineering planning for the next phase of the effort, manufacturing considerations, and the planning for production engineering. *See also:* **critical design review; preliminary design review.** [610.12]

system development cycle. The period of time that begins with the decision to develop a system and ends when the system is delivered to its end user. *Note:* This term is sometimes used to mean a longer period of time, either the period that ends when the system is no longer being enhanced, or the entire system life cycle. *Contrast with:* **system life cycle.** *See also:* **software development cycle.** [610.12]

system flowchart (flow chart). *See:* **flowchart.** [610.12]

system library. A software library containing system-resident software that can be accessed for use or incorporated into other programs by reference; for example, a macro library. *Contrast with:* **master library; production library; software development library; software repository.** [610.12]

system life cycle. The period of time that begins when a system is conceived and ends when the system is no longer available for use. *See also:* **system development cycle; software life cycle.** [610.12]

system model. In computer performance evaluation, a representation of a system depicting the relationships between workloads and performance measures in the system. *See also:* **workload model.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

system profile. A set of measurements used in computer performance evaluation, describing the proportion of time each of the major resources in a computer system is busy, divided by the time that resource is available. [610.12]

system requirements review (SRR). A review conducted to evaluate the completeness and adequacy of the requirements defined for a system; to evaluate the system engineering process that produced those requirements; to assess the results of system engineering studies; and to evaluate system engineering plans. *See also:* **software requirements review.** [610.12]

system resources chart. *See:* **block diagram.** [610.12]

system software. Software designed to facilitate the operation and maintenance of a computer system and its associated programs; for example, operating systems, assemblers, utilities. *Contrast with:* **application software.** *See also:* **support software.** [610.12]

system state. *See:* **state.** [610.3]

system testing. Testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. *See also:* **component testing; integration testing; interface testing; unit testing.** [610.12]

table. A two-dimensional array. *See also:* **code-decode table.** (See below for an example.) [610.5]

State	Abbreviation	Zone
Alabama	AL	2
Alaska	AK	9
...		
West Virginia	WV	3
Wisconsin	WI	4
Wyoming	WY	3

Fig 68
Table

table lookup (TLU). (1) A procedure for obtaining the value of a function corresponding to a given argument from a table of function values. *See also:* **look-up table.** [610.1]
(2) The process of obtaining the value y corresponding to an argument x from a two-dimensional table of (x,y) pairs. *See also:* **associative lookup; direct lookup.** [610.5]

tabular model. A symbolic model whose properties are expressed in tabular form; for example, a truth table that represents the logic of an OR gate. *Contrast with:* **graphical model; mathematical model; narrative model; software model.** [610.3]

tabulation character. A format effector character that causes the print or display position to move to the next corresponding horizontal or vertical position in a series of predetermined positions. *See also:* **horizontal tabulation character; vertical tabulation character.** [610.5]

tag. One or more characters associated with a set of data, containing information about the set. [610.5]

tag sort. A sort that uses the address table sorting technique. [610.5]

tape merge sort. An external merge sort in which the auxiliary storage used is a magnetic tape. *See also:* **direct-access merge sort.** [610.5]

target. In micrographics, any document or chart containing identification information or a resolution test chart. *See also:* **flash card.** [610.2]

target language. The language in which the output from a machine-aided translation process is represented. For example, the language output by an assembler or compiler. *Syn:* **object language.** *Contrast with:* **source language.** [610.2a, 610.12]

target machine. (1) The computer on which a program is intended to execute. *Contrast with:* **host machine (1).** [610.12]
(2) A computer being emulated by another computer. *Contrast with:* **host machine (2).** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052938 2

target program. *See:* **object program.** [610.12]

task. (1) A sequence of instructions treated as a basic unit of work by the supervisory program of an operating system. [610.12]
(2) In software design, a software component that can operate in parallel with other software components. [610.12]

taxonomy. A scheme that partitions a body of knowledge and defines the relationships among the pieces. It is used for classifying and understanding the body of knowledge. [610.12]

technical management. The application of technical and administrative resources to plan, organize, and control engineering functions. [610.12]

technical standard. A standard that describes the characteristics of applying accumulated technical or management skills and methods in the creation of a product or performing a service. [610.12]

techniques. Technical and managerial procedures that aid in the evaluation and improvement of the software development process. [610.12]

teleautograph. A telegraphic writing instrument, in which movement of a pen at the transmitting end causes corresponding movement of a pen at the remote receiving instrument. *Syn:* **telewriter.** [610.2]

telecommunication. The transmission of signals over long distance, such as by telegraph, radio, or television. *See also:* **computer conferencing; office automation.** [610.2]

telecommuting. An employment alternative involving working at home using a computer and telecommunication system instead of commuting between home and workplace. [610.2]

teleconferencing. A form of communication that uses telephones, computer networks, and television to allow participants at different geographic locations to confer. [610.2]

telecopier. A device used for facsimile transmission. [610.2]

telefax. *Syn:* **facsimile transmission.** [610.2]

teleinformatics. Data transfer via telecommunication systems. [610.2]

teleordering. Use of a telecommunication system to accept orders from customers at remote locations. *Syn:* **on-line ordering; teleshopping.** [610.2]

telephotography. *Syn:* **facsimile telegraphy.** [610.2]

teleprocessing. (1) *See:* **remote-access data processing.** [610.2]
(2) *See:* **distributed data processing.*** [610.2]
*Deprecated.

telereference. The use of a telecommunication system to reference data at some remote location. [610.2]

teleshopping. *Syn:* **teleordering.** [610.2]

teletext. A form of videotex that allows users to receive textual or pictorial material via broadcast signals interpreted by a special decoder attached to a television set. *Contrast with:* **viewdata.** [610.2]

teletype exchange (telex or TEX). A service that permits the transmission of data using commercial telecommunication facilities. [610.2]

teletypesetting (TTS). Use of a telecommunication system to allow typesetting to be done at remote locations. [610.2]

telewriter. *Syn:* **teleautograph.** [610.2]

telex. Acronym for **teletype exchange.** [610.2]

template matching. (1) An image processing technique in which patterns or shapes are detected by comparison with prespecified patterns or shapes called templates. *Syn:* **matched filtering.** *See also:* **image matching.** [610.4]
(2) A pattern recognition technique using the principle described in (1). [610.4]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

temporal cohesion. A type of cohesion in which the tasks performed by a software module are all required at a particular phase of program execution; for example, a module containing all of a program's initialization tasks. *Contrast with:* **coincidental cohesion; communicational cohesion; functional cohesion; logical cohesion; procedural cohesion; sequential cohesion.** [610.12]

tens complement. The radix complement of a decimal numeral, which may be formed by subtracting each digit from 9, then adding 1 to the least significant digit and executing any required carries. For example, the tens complement of 4830 is 5170. *Syn:* **complement on ten.** [610.1]

terdenary. (1) Pertaining to a selection in which there are 13 possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of 13. [610.1]

terminal node. In a tree, a node that has no subtrees. *Syn:* **external node; end point.** *Contrast with:* **nonterminal node.** *See also:* **leaf; root node.** [610.5]

termination construct. A program construct that results in a halt or exit. [610.12]

terminology bank. *Syn:* **automated glossary.** [610.2]

ternary. (1) Pertaining to a selection in which there are three possible outcomes. [610.1]
(2) Pertaining to the numeration system with a radix of 3. [610.1]

ternary incremental representation. Representation of changes in variables in which the value of an increment is plus one, zero, or minus one. *Syn:* **incremental ternary representation.** [610.1]

ternary relation. A relation with three attributes. [610.5]

test. (1) An activity in which a system or component is executed under specified conditions, the results are observed or recorded, and an evaluation is made of some aspect of the system or component. [610.12]
(2) To conduct an activity as in (1). [610.12]
(3) A set of one or more test cases. [610.12]
(4) A set of one or more test procedures. [610.12]
(5) A set of one or more test cases and procedures. [610.12]

testability. (1) The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met. [610.12]
(2) The degree to which a requirement is stated in terms that permit establishment of test criteria and performance of tests to determine whether those criteria have been met. [610.12]

test bed. An environment containing the hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test. [610.12]

test case. (1) A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. [610.12]
(2) Documentation specifying inputs, predicted results, and a set of execution conditions for a test item. *See also:* **test case generator; test case specification.** [610.12]

test case generator. A software tool that accepts as input source code, test criteria, specifications, or data structure definitions; uses these inputs to generate test input data; and, sometimes, determines expected results. *Syn:* **test data generator; test generator.** [610.12]

test case specification. A document that specifies the test inputs, execution conditions, and predicted results for an item to be tested. *Syn:* **test description; test specification.** *See also:* **test incident report; test item transmittal report; test log; test plan; test procedure; test report.** [610.12]

test coverage. The degree to which a given test or set of tests addresses all specified requirements for a given system or component. [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

test criteria. The criteria that a system or component must meet in order to pass a given test. *See also:* **acceptance criteria; pass-fail criteria.** [610.12]

test data generator. *See:* **test case generator.** [610.12]

test description. *See:* **test case specification.** [610.12]

test design. Documentation specifying the details of the test approach for a software feature or combination of software features and identifying the associated tests. [610.12]

test documentation. Documentation describing plans for, or results of, the testing of a system or component. Types include test case specification, test incident report, test log, test plan, test procedure, test report. [610.12]

test driver. A software module used to invoke a module under test and, often, provide test inputs, control and monitor execution, and report test results. *Syn:* **test harness.** [610.12]

test generator. *See:* **test case generator.** [610.12]

test harness. *See:* **test driver.** [610.12]

test incident report. A document that describes an event that occurred during testing which requires further investigation. *See also:* **test case specification; test item transmittal report; test log; test plan; test procedure; test report.** [610.12]

testing. (1) The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component. [610.12]
(2) The process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs) and to evaluate the features of the software items. [610.12]
See also: **acceptance testing; benchmark; checkout; component testing; development testing; dynamic analysis; formal testing; functional testing; informal testing; integration testing; interface testing; loopback testing; mutation testing; operational testing; performance testing; qualification testing; regression testing; stress testing; structural testing; system testing; unit testing.** [610.12]

test item. A software item which is an object of testing. [610.12]

test item transmittal report. A document that identifies one or more items submitted for testing. It contains current status and location information. *See also:* **test case specification; test incident report; test log; test plan; test procedure; test report.** [610.12]

test log. A chronological record of all relevant details about the execution of a test. *See also:* **test case specification; test incident report; test item transmittal report; test plan; test procedure; test report.** [610.12]

test objective. An identified set of software features to be measured under specified conditions by comparing actual behavior with the required behavior described in the software documentation. [610.12]

test phase. The period of time in the software life cycle during which the components of a software product are evaluated and integrated, and the software product is evaluated to determine whether or not requirements have been satisfied. [610.12]

test plan. (1) A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning. [610.12]
(2) A document that describes the technical and management approach to be followed for testing a system or component. Typical contents identify the items to be tested, tasks to be performed, responsibilities, schedules and required resources for the testing activity. [610.12]
See also: **test case specification; test incident report; test item transmittal report; test log; test procedure; test report.** [610.12]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

test procedure. (1) Detailed instructions for the set-up, execution, and evaluation of results for a given test case. [610.12]
(2) A document containing a set of associated instructions as in (1). [610.12]
(3) Documentation specifying a sequence of actions for the execution of a test. [610.12]
Syn: **test procedure specification; test script.** *See also:* **test case specification; test incident report; test item transmittal report; test log; test plan; test report.** [610.12]

test procedure specification. *See:* **test procedure.** [610.12]

test readiness review (TRR). (1) A review conducted to evaluate preliminary test results for one or more configuration items; to verify that the test procedures for each configuration item are complete, comply with test plans and descriptions, and satisfy test requirements; and to verify that a project is prepared to proceed to formal testing of the configuration items. [610.12]
(2) A review as in (1) for any hardware or software component. [610.12]
Contrast with: **code review; formal qualification review; design review; requirements review.** [610.12]

test repeatability. An attribute of a test, indicating that the same results are produced each time the test is conducted. [610.12]

test report. A document that describes the conduct and results of the testing carried out for a system or component. *Syn:* **test summary report.** *See also:* **test case specification; test incident report; test item transmittal report; test log; test plan; test procedure.** [610.12]

test script. *See:* **test procedure.** [610.12]

test set architecture. The nested relationships between sets of test cases that directly reflect the hierarchic decomposition of the test objectives. [610.12]

test specification. *See:* **test case specification.** [610.12]

test summary report. A document summarizing testing activities and results. It also contains an evaluation of the corresponding test items. *See also:* **test case specification; test incident report; test item transmittal report; test log; test plan; test procedure; test report.** [610.12]

test unit. A set of one or more computer program modules together with associated control data, (for example, tables), usage procedures, and operating procedures that satisfy the following conditions: (a) All modules are from a single computer program; (b) At least one of the new or changed modules in the set has not completed the unit test; (c) The set of modules together with its associated data and procedures are the sole object of a testing process. [610.12]

tetrad. A group of four closely related items or digits. [610.1]

TEX. (1) Acronym for **teletype exchange.** [610.2]
(2) A public-domain word processing language. [610.2]

text. In word processing, information that is intended for presentation for human comprehension in a two-dimensional form. Text may consist of symbols, phrases, sentences in natural or artificial language, pictures, diagrams, and tables. [610.2]

text editing. The process of entering, altering, and viewing text. [610.2]

text editor. A computer program, often part of a word processing system, that allows a user to enter, alter, and view text. *See also:* **document editor; full-screen editor; line editor; program editor.** *Syn:* **editor.** [610.2, 610.12]

text end adjustment. The ability of a text formatter to automatically reformat text to comply with specified line lengths and page sizes. *See also:* **adjust line mode.** [610.2]

text formatter. A computer program, often part of a word processing system, that interprets formatting commands embedded in text and performs the indentation, pagination, tabulation, underscoring, and other formatting procedures indicated by the commands. *Syn:* **print formatter.** [610.2]

text formatting. In word processing, the process of interpreting formatting commands embedded in text and performing the inden-

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

tation, pagination, tabulation, underscoring, and other formatting procedures indicated by the formatting commands. *Syn:* **print formatter.** [610.2]

text processing. *See:* **word processing.** [610.2]

texture. In image processing, an attribute representing the spatial arrangement of the gray levels of the pixels in a region. [610.4]

think time. The elapsed time between the end of a prompt or message generated by an interactive system and the beginning of a human user's response. *See also:* **port-to-port time; response time; turnaround time.** [610.12]

thinning. An image processing technique in which regions are reduced to sets of thin curves. [610.4]

third generation language (3GL). *See:* **high-order language.** [610.12]

third normal form (3NF). One of the forms used to characterize relations; a relation is said to be in third normal form if it is in second normal form and if no nonprime attribute is transitively dependent on the primary key. *See also:* **Boyce/Codd normal form.** [610.5]

SECOND NORMAL FORM

ORDER2 = {ORDER-NO} + DATE + CUSTOMER-NO
+ CUSTOMER-NAME
+ CUSTOMER-ADDRESS
+ TOTAL-ORDER-AMOUNT
ORDER-ITEM2 = {ORDER-NO + ITEM-NO}
+ QUANTITY-ORDERED
+ EXTENDED-PRICE
ITEM2 = {ITEM-NO} + ITEM-DESCRIPTION
+ UNIT-PRICE

THIRD NORMAL FORM

ORDER3 = {ORDER-NO} + DATE + CUSTOMER-NO
+ TOTAL-ORDER-AMOUNT
CUSTOMER3 = {CUSTOMER-NO}
+ CUSTOMER-NAME
+ CUSTOMER-ADDRESS
ORDER-ITEM3 = {ORDER-NO + ITEM-NO}
+ QUANTITY-ORDERED + EXTENDED-PRICE
ITEM3 = {ITEM-NO} + ITEM-DESCRIPTION
+ UNIT-PRICE

Note: In second normal form, nonprime attributes CUSTOMER-NAME and CUSTOMER-ADDRESS are transitively dependent on CUSTOMER-NO. Keys shown in brackets.

Fig 69
Third Normal Form

thrashing. A state in which a computer system is expending most or all of its resources on overhead operations, such as swapping data between main and auxiliary storage, rather than on intended computing functions. [610.12]

thread. In a tree, a set of link fields, one in each node, each of which points to the successor or predecessor of that node with respect to a particular traversal order. [610.5]

threaded tree. A tree whose nodes contain link fields for one or more threads, allowing nonrecursive traversal of the tree. *See also:* **doubly-threaded tree; left-threaded three; right-threaded tree; triply-threaded tree.** [610.5]

three-address instruction. A computer instruction that contains three address fields. For example, an instruction to add the contents of locations A and B, and place the results in location C. *Contrast with:* **one-address instruction; two-address instruction; four-address instruction; zero-address instruction.** [610.12]

three-bit byte. *See:* **triplet.** [610.5]

three-plus-one address instruction. A computer instruction that contains four address fields, the fourth containing the address of the instruction to be executed next. For example, an instruction to add the contents of locations A and B, place the results in location C, then execute the instruction at location D. *Contrast with:* **one-plus-one address instruction; two-plus-one address instruction; four-plus-one address instruction.** [610.12]

threshold. (1) A logic operator having the property that if P is a statement, Q is a statement, R is a statement,..., then the threshold of P, Q, R,... is true if at least N statements are true, false if less than N statements are true, where N is a specified non-negative integer called the threshold condition. [610.1]
(2) The threshold condition as in (1). [610.1]
(3) In image processing, a specified gray level used for producing a binary image. *See also:* **thresholding.** [610.4]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

thresholding. The process of producing a binary image from a gray scale image by assigning each output pixel the value 1 if its corresponding input pixel is at or above a specified gray level (the threshold) and the value 0 if the input pixel is below that threshold. [610.4]

throughput. The amount of work that can be performed by a computer system or component in a given period of time; for example, number of jobs per day. *See also:* **turnaround time; workload model.** [610.12]

tier chart. *See:* **call graph.** [610.12]

time-dependent event. An event that occurs at a predetermined point in time or after a predetermined period of time has elapsed. *See also:* **conditional event.** [610.3]

time-interval simulation. *See:* **time-slice simulation.** [610.3]

time out. (1) A condition that occurs when a predetermined amount of time elapses without the occurrence of an expected event. For example, the condition that causes termination of an on-line process if no user input is received within a specified period of time. [610.12]
(2) To experience the condition in (1).[610.12]

time sharing. A mode of operation that permits two or more users to execute computer programs concurrently on the same computer system by interleaving the execution of their program. *Note:* Time sharing may be implemented by time slicing, priority-based interrupts, or other scheduling methods. [610.12]

time-slice simulation. (1) A discrete simulation that is terminated after a specific amount of time has elapsed; for example, a model depicting the year-by-year forces affecting a volcanic eruption over a period of 100 000 years. *Syn:* **time-interval simulation.** *See also:* **critical event simulation.** [610.3]
(2) A discrete simulation of continuous events in which time advances by intervals chosen independent of the simulated events; for example, a model of a time multiplexed communication system with multiple channels transmitting signals over a single transmission line in very rapid succession. [610.3]

time slicing. A mode of operation in which two or more processes are each assigned a small, fixed amount of continuous processing time on the same processor, and the processes execute in a round-robin manner, each for its allotted time, until all are completed. [610.12]

time variable. A variable whose value represents simulated time or the state of the simulation clock. [610.3]

timing. The process of estimating or measuring the amount of execution time required for a software system or component. *Contrast with:* **sizing.** [610.12]

timing analyzer. A software tool that estimates or measures the execution time of a computer program or portion of a computer program, either by summing the execution times of the instructions along specified paths or by inserting probes at specified points in the program and measuring the execution time between probes. [610.12]

TLU. Acronym for **table lookup.** [610.5]

top. In a queue or a stack, the position of the next item to be retrieved. *Contrast with:* **bottom.** [610.5]

top-down. Pertaining to an activity that starts with the highest level component of a hierarchy and proceeds through progressively lower levels; for example, top-down design; top-down testing. *Contrast with:* **bottom-up.** *See also:* **critical piece first.** [610.12]

total correctness. In proof of correctness, a designation indicating that a program's output assertions follow logically from its input assertions and processing steps, and that, in addition, the program terminates under all specified input conditions. *Contrast with:* **partial correctness.** [610.12]

tournament sort. A repeated selection sort in which each of the subsets that make up the set

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

to be sorted consists of no more than two items. [610.5]

trace. (1) A record of the execution of a computer program, showing the sequence of instructions executed, the names and values of variables, or both. Types include execution trace, retrospective trace, subroutine trace, symbolic trace, variable trace. [610.12]
(2) To produce a record as in (1). [610.12]
(3) To establish a relationship between two or more products of the development process; for example, to establish the relationship between a given requirement and the design element that implements that requirement. [610.12]

traceability. (1) The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another; for example, the degree to which the requirements and design of a given software component match. *See also:* **consistency.** [610.12]
(2) The degree to which each element in a software development product establishes its reason for existing; for example, the degree to which each element in a bubble chart references the requirement that it satisfies. [610.12]

traceability matrix. A matrix that records the relationship between two or more products of the development process; for example, a matrix that records the relationship between the requirements and the design of a given software component. [610.12]

tracking. An image segmentation technique in which arcs are detected by searching sequentially from one arc pixel to the next. [610.4]

trailer. Identification or control information placed at the end of a file or message. *Contrast with:* **header (2).** [610.12]

trailing decision. A loop control that is executed after the loop body. *Contrast with:* **leading decision.** *See also:* **UNTIL.** [610.12]

trailing zero. A zero that comes after the last digit in a numeric representation that is non-zero, and that is to the right of the decimal point; for example, the two zeros in "324.600." *Contrast with:* **leading zero.** [610.5]

transaction. (1) An event that requires data contained in a master file to be processed. *See also:* **add transaction; change transaction; delete transaction; null transaction; update transaction.** [610.2]
(2) In software engineering, a data element, control element, signal, event, or change of state that causes, triggers, or initiates an action or sequence of actions. [610.12]

transaction analysis. A software development technique in which the structure of a system is derived from analyzing the transactions that the system is required to process. *Syn:* **transaction-centered design.** *See also:* **data structure-centered design; input-process-output; modular decomposition; object-oriented design; rapid prototyping; stepwise refinement; structured design; transform analysis.** [610.12]

transaction-centered design. *See:* **transaction analysis.** [610.12]

transaction code. An identifier associated with a transaction and representing the operation to be carried out by that transaction. For example, "A" for an add transaction, "D" for a delete transaction. [610.2]

transaction file. An organized collection of transaction records. *Syn:* **detail file.** *Contrast with:* **master file.** [610.2]

transaction matrix. A matrix that identifies possible requests for database access and relates each request to information categories or elements in the database. [610.12]

transaction record. A record, representing one transaction, used to process data stored in a master file. *See also:* **add transaction; change transaction; delete transaction; null transaction; update transaction.** [610.2]

transfer. (1) To send data from one place and receive it at another. *See also:* **transmit.** [610.5a, 610.12]
(2) To relinquish control by one process and assume it at another, either with expectation

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

of return (*see:* **call**) or without such expectation (*see:* **jump**). [610.12]

transform analysis. A software development technique in which the structure of a system is derived from analyzing the flow of data through the system and the transformations that must be performed on the data. *Syn:* **transformation analysis; transform-centered design.** *See also:* **data structure-centered design; input-process-output; modular decomposition; object-oriented design; rapid prototyping; stepwise refinement; structured design; transaction analysis.** [610.12]

transformation analysis. *See:* **transform analysis.** [610.12]

transform-centered design. *See:* **transform analysis.** [610.12]

transient error. An error that occurs once, or at unpredictable intervals. *See also:* **intermittent fault; random failure.** [610.12]

transient read. *See*: **dirty read.** [610.5]

transitive dependency. A type of dependency among attributes in a relation, in which a nonprime attribute A is said to be transitively dependent on another attribute B if and only if there is another attribute C that is functionally dependent on B and functionally determining A but not B. *Contrast with*: **nontransitive dependency.** [610.5]

translate. To transform data from one language to another. [610.5]

translator. A computer program that transforms a sequence of statements expressed in one language into an equivalent sequence of statements expressed in another language. *See also:* **assembler; compiler.** [610.12]

transliterate. To convert data character-by-character from one character set to another. [610.5]

transmission control character. (1) Any control character used to control or facilitate transmission of data. [610.5]
(2) Any character transmitted that is not part of the message being transferred, but that is used to control or to facilitate the transfer. *Syn*: **communication control character.** [610.5]

transmit. To send data from one place for reception elsewhere. *See also*: **transfer.** [610.5]

transparent. (1) In data transmission, pertaining to information that does not contain transmission control characters. [610.5]
(2) To perform in a manner that is invisible to, and of no concern to a user. For example, a computer program may perform file allocation, database operations, and housekeeping operations **transparent** to its user. [610.5]

transportability. *See:* **portability.** [610.12]

transportable computer. A personal computer that weighs more than 21 pounds, yet is designed and configured to permit easy transportation. *See also:* **portable computer.**[610.2]

transposed file. A file in which corresponding fields in corresponding records are stored contiguously, in contrast to the usual practice of storing entire records contiguously. [610.5]

trap. (1) A conditional jump to an exception or interrupt handling routine, often automatically activated by hardware, with the location from which the jump occurred recorded.
(2) To perform the operation in (1). [610.12]

traversal. The process of enumerating or visiting each of the nodes of an ordered tree exactly once. *See also*: **converse inorder traversal; converse preorder traversal; converse postorder traversal; inorder traversal; preorder traversal; postorder traversal; traverse; traversal order.** [610.5]

traversal order. The order in which the nodes of a tree are visited in a traversal. [610.5]

traverse. To enumerate or to visit each of the nodes of an ordered tree exactly once. *See also*: **traversal.** [610.5]

tree. A nonlinear data structure consisting of

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

a finite set of nodes in which one node is called the root node and the remaining nodes are partitioned into disjoint sets, called subtrees, each of which is itself a tree. *Note*: The nodes are connected by pointers. *Syn*: **rooted tree; tree structure.** *See also*: **height-balanced tree; null tree; n-ary tree; *n-m* tree; ordered tree; search tree; subtree; threaded tree; trie; unordered tree.** [610.5]

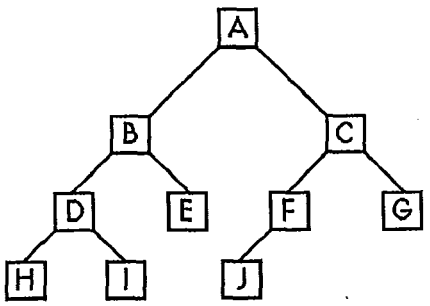


Fig 70
Tree

tree insertion sort. An insertion sort in which the items in the set to be sorted are treated as nodes on a tree. *Contrast with*: **tree selection sort.** [610.5]

tree selection sort. A selection sort in which the items in the set to be sorted are treated as nodes on a tree. *See also*: **quadratic selection sort.** *Contrast with*: **tree insertion sort.** [610.5]

tree structure. *See*: **tree.** [610.5]

triad. A group of three closely related items or digits. [610.1]

trie. An *n*-ary tree each of whose nonterminal nodes is the parent of a sequence of subtrees, where the *k*-th subtree represents the *k*-th digit or character in an *n*-character alphabet. *Note 1*: A sequence of nodes (length *p*) from the root of a trie to the root of a subtree represents the first *p* digits or characters of the keys of the elements represented by that subtree. *Note 2*: The term is pronounced "try" and is derived from the word "re-trie-val." *See also*: **binary radix trie search; multiway radix trie search; radix trie search.** [610.5]

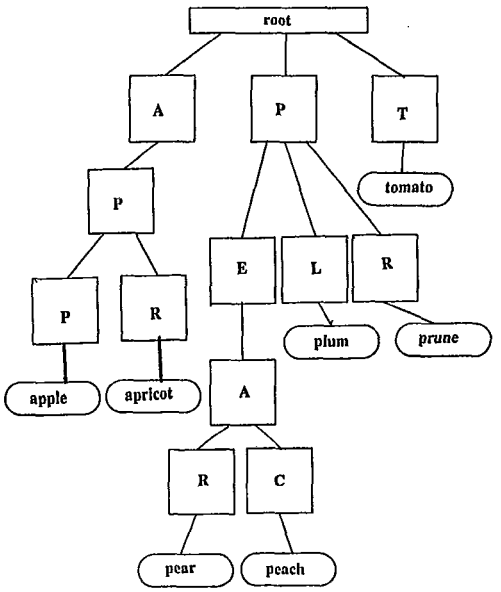


Fig 71
Trie

triple precision. Pertaining to the use of three computer words to represent a number in order to preserve or gain precision. *Contrast with:* **double precision; single precision.** *See also:* **multiple precision.** [610.1, 610.5]

triple-precision arithmetic. Computer arithmetic performed with operands that are expressed in triple-precision representation. [610.1]

triplet. A group of three adjacent digits operated upon as a unit. *Syn*: **three-bit byte.** [610.1, 610.5]

triply-threaded tree. A binary tree in which each node contains three link fields: one for its parent node and one for each of its left child and right child nodes. *Contrast with*: **doubly-threaded tree.** [610.5]

TRR. Acronym for **test readiness review.** [610.12]

true complement. *See:* **radix complement.** [610.1]

truncate. (1) *See:* **round down.** [610.1]
(2) To terminate a computational process in accordance with some rule; for example, to end the evaluation of a power series at a specified term. [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

(3) To remove the beginning or ending entities in a string; for example, the string 'PINEAPPLE,' when truncated on the right to six characters, is 'PINEAP.' [610.5]
(4) To delete or omit one or more of the digits in a representation of a number; for example, the numbers 57.5634 and 25.437, when truncated to two decimal digits, become 57.56 and 25.43. *Contrast with:* **round.** [610.5]

truncation error. An error caused by truncation. [610.1]

truncation. The process of truncating. [610.5]

truth function. A function that may take one of two possible values: true or false. [610.1]

truth table. An operation table that describes a truth function by listing all possible combinations of input values and giving the corresponding output values. *Syn:* **Boolean operation table.** *See also:* **AND; NOT; OR.** [610.1]

TTS. Acronym for **teletypesetting.** [610.2]

tuple. (1) A suffix meaning "an ordered set of items," as in *n*-tuple. [610.5]
(2) In a relational data model, a set of values of related attributes. *Note*: Often thought of as a row in a table. *Syn*: **row.** *See also*: **attribute; relation.** [610.5]

turnaround time. The elapsed time between the submission of a job to a batch processing system and the return of completed output. *See also:* **port-to-port time; response time; think time.** [610.12]

turning center. A numerical control machine capable of performing lathe-oriented operations, such as boring, facing, turning, and threading. [610.2]

turnkey. Pertaining to a hardware or software system delivered in a complete, operational state. [610.12]

tutorial simulation. *See:* **instructional simulation.** [610.3]

twin segment. In a hierarchical database, a child segment N that shares a common parent segment with another child segment M. Segments N and M are said to be twin segments. *See also:* **logical twin segment; physical twin segment.** [610.5]

two-address instruction. A computer instruction that contains two address fields. For example, an instruction to add the contents of A to the contents of B. *Syn:* **double-operand instruction.** *Contrast with:* **one-address instruction; three-address instruction; four-address instruction; zero-address instruction.** [610.12]

two-level address. An indirect address that specifies the storage location containing the address of the desired operand. *See also:* ***n*-level address.** [610.12]

two-level encoding. A microprogramming technique in which different microoperations may be encoded identically into the same field of a microinstruction, and the one that is executed depends upon the value in another field internal or external to the microinstruction. *See also:* **bit steering; residual control.** *Contrast with:* **single-level encoding.** [610.12]

two-out-of-five code. A BCD code in which each decimal digit is represented by a five-digit numeral of which two bits are in one state (usually ones) and three are in the other state. *See also:* ***m*-out-of-*n* code.** [610.1]

two-plus-one address instruction. A computer instruction that contains three address fields, the third containing the address of the instruction to be executed next. For example, an instruction to add the contents of A to the contents of B, then execute the instruction at location C. *Contrast with:* **one-plus-one address instruction; three-plus-one address instruction; four-plus-one address instruction.** [610.12]

two-scale. *See:* **binary notation.** [610.1]

twos complement. The radix complement of a binary numeral, which may be formed by subtracting each digit from 1, then adding 1 to the least significant digit and executing any required carries. For example, the twos complement of 1101 is 0011. *Syn:* **complement on two.** [610.1]

twos-complement arithmetic. Computer arithmetic performed with operands that are expressed in twos-complement notation. [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

twos-complement notation. A binary numeration system in which negative numbers are represented by their twos complement and positive numbers are expressed in their usual binary form. *Contrast with:* **sign-magnitude notation.** [610.1]

two-state variable. *See:* **binary variable.** [610.1]

two-valued variable. *See:* **binary variable.** [610.1]

two-way chain. *See:* **doubly linked list.** [610.5]

two-way insertion sort. An insertion sort in which each item in the set to be sorted is inserted in its proper position in the sorted set such that the first item is placed in the middle of the output set and space is made for subsequent items by moving the previously-inserted items to the right or left. *Contrast with:* **binary insertion sort.** [610.5]

two-way merge sort. A merge sort in which the set to be sorted is divided into two subsets, the items in each subset are sorted, and the subsets are merged by comparing the smallest items of each subset, outputting the smallest of those, then repeating the process. *See also:* **multiway merge sort; natural two-way merge sort; straight two-way merge sort.** [610.5]

type. *See:* **data type.** [610.12]

type font. *Syn:* **character font.** [610.2]

type I error. *See:* **misidentification.** [610.4]

type II error. *See:* **false identification.** [610.4]

UDF. Acronym for **unit development folder.** *See:* **software development file.** [610.12]

ultrafiche. In micrographics, microfiche with images reduced more than ninety times. [610.2]

unary operation. *See:* **monadic operation.** [610.1]

unary operator. *See:* **monadic operator.** [610.1]

unary relation. A relation with one attribute. [610.5]

unbalanced error. (1) A set of error values in which the maximum and minimum are not necessarily opposite in sign and equal in magnitude. [610.1]
(2) A set of error values whose average is not zero. *Contrast with:* **balanced error.** [610.1]

unbalanced merge. A merge in which the subsets to be merged are unequally distributed among half of the available auxiliary storage devices, then the subsets are merged onto the other half of the auxiliary storage devices. *Contrast with:* **balanced merge.** [610.5]

unbalanced merge sort. A merge sort in which the sorted subsets created by internal sorts are unequally distributed among some of the available storage, the subsets are merged onto the remaining available storage, and this process is repeated until all the items are in one sorted set. *Contrast with:* **balanced merge sort.** *See also:* **polyphase merge sort.** [610.5]

unbiased rounding. A rounding process in which the rules for adjusting the retained numeral ensure that the average rounding error is zero. [610.1]

unblock.* *See:* **deblock.** [610.5]
* Deprecated.

unblocked record. A record that is contained in exactly one entire block. *See also:* **blocked record; spanned record.** [610.5]

uncertainty. The upper bound on an absolute error or relative error. [610.1]

unconditional branch.* *See:* **unconditional jump.** [610.12]
*Deprecated.

unconditional jump. A jump that takes place regardless of execution conditions. *Contrast with:* **conditional jump.** [610.12]

underflow. The condition that arises when the result of a floating-point arithmetic operation is smaller than the smallest non-zero number that can be represented in a digital computer. *Syn:* **arithmetic underflow.** [610.1]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

underflow error. The error caused by an underflow condition in computer arithmetic. [610.1]

underflow exception. An exception that occurs when the result of an arithmetic operation is too small a fraction to be represented by the storage location designated to receive it. *See also:* **addressing exception; data exception; operation exception; overflow exception; protection exception.** [610.12]

undirected graph. A graph (2) in which no direction is implied in the internode connections. *Contrast with:* **directed graph.** [610.5a, 610.12a]

uniform probing. Open-address hashing in which collision resolution is handled by selecting positions at uniform distances from the original position in the hash table until an available position is found. *Contrast with:* **linear probing; quadratic probing; random probing.** [610.5]

uniform random number. Any member of a random number sequence that has a uniform statistical distribution. [610.1]

union. (1) A relational operator that combines two relations of the same degree and results in a relation containing all of the tuples that are in either of the original relations. *See also:* **difference; intersection; join; product; projection; selection.** [610.5]

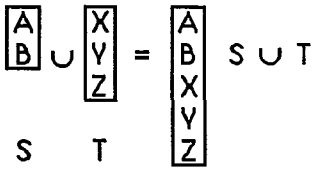


Fig 72
Union

(2) *See:* **OR.** [610.1]

unit. (1) A separately testable element specified in the design of a computer software component. [610.12]
(2) A logically separable part of a computer program. [610.12]
(3) A software component that is not subdivided into other components. [610.12]
(4) *See:* **test unit.** *Note:* The terms "module," "component," and "unit" are often used interchangeably or defined to be sub-elements of one another in different ways depending upon the context. The relationship of these terms is not yet standardized. [610.12]

unitary code. A code having only one digit; the number of times it is repeated determines the quantity it represents. [610.1]

unit development folder (UDF). *See:* **software development file.** [610.12]

unit-distance code. A code in which the Hamming distance between consecutive numerals is 1. *Syn:* **continuous-progression code; cyclic permuted code.** [610.1]

uniterm indexing. A variation of derivative indexing in which each keyword must be a single word. [610.2]

unit requirements documentation. Documentation that sets forth the functional, interface, performance, and design constraint requirements for a test unit. [610.12]

units position. In a positional notation system, the position corresponding to the zero power of the radix. This is the right-most position in a numeral representing an integer.[610.1]

unit string. A string consisting of only one entity. [610.5]

unit testing. Testing of individual hardware or software units or groups of related units. *See also:* **component testing; integration testing; interface testing; system testing.** [610.12]

universal product code (UPC). A bar code appearing on many retail products to uniquely identify the product. The code is designed to be read by an optical scanner attached to an electronic cash register. [610.2]



Fig 73
Example of Universal Product Code

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

unnormalized form. The form assumed by data that have not been normalized. *Contrast with*: **normalized form.** [610.5]

unnormalized relation. A relation that is not in normal form. *Contrast with*: **normalized relation.** [610.5]

unodecimal. (1) Pertaining to a selection in which there are 11 possible outcomes. [610.1] (2) Pertaining to the numeration system with a radix of 11. [610.1]

unordered list. A list in which data items are not arranged in any specific order. *Contrast with*: **ordered list.** *Syn*: **random-ordered list.** [610.5]

unordered tree. A tree in which the left-to-right order of the subtrees of a given node is not significant. *Contrast with*: **ordered tree.** [610.5]

unpack. To recover the original form of one or more data items from packed form. *Contrast with*: **pack.** [610.5, 610.12]

unpacked decimal data. *See*: **zoned decimal data.** [610.5]

unsharp masking. In image processing, a sharpening technique in which an intentionally blurred version of the image is subtracted from the image itself. [610.4]

unsigned packed decimal data. Integer data in which each decimal digit is represented in binary, occupying four bits. *Note*: Since no sign is stored, only non-negative integers can be represented. *See also*: **packed decimal data.** [610.5]

decimal	75_{10}
unsigned packed decimal	$0111\ 0101_2$

unstratified language. A language that can be used as its own metalanguage; for example, English, German. *Contrast with:* **stratified language.** [610.12]

UNTIL. A single-entry, single-exit loop, in which the loop control is executed after the loop body. *Syn:* **post-tested iteration.** *Contrast with:* **closed loop; WHILE.** *See also:* **trailing decision.** [610.12]

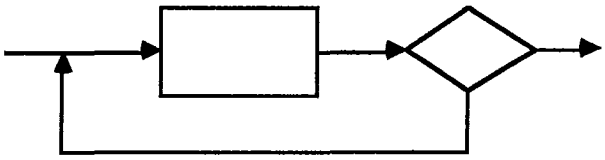


Fig 74
UNTIL Construct

unwind. In programming, to state explicitly and in full all of the instructions involved in multiple executions of a loop. *See also:* **straightline coding.** [610.12]

up. Pertaining to a system or component that is operational and in service. Such a system is either busy or idle. *Contrast with:* **down.** *See also:* **busy; idle.** [610.12]

UPC. Abbreviation for **universal product code.** [610.2]

update. (1) To change information in accordance with information that is more recent than that which was available previously. For example, a master file containing account balances might be updated nightly to reflect transactions precessed the previous day. [610.5] (2) To replace data in a storage device or on a data medium. *See also*: **delete; read; write.** [610.5]

update access. A type of access to data in which the data can be updated. *See also*: **delete access; read-only access; read/write access; write access.** [610.5]

update transaction. A transaction that modifies a master file by adding, deleting, or changing data to make it more current. *See also:* **add transaction; change transaction; delete transaction; null transaction.** [610.2]

updateable microfilm. Microfilm that permits the addition or deletion of images. [610.2]

upload. (1) To transfer some collection of data from some storage location to a computer memory. [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

(2) To transfer some collection of data from the memory of a small computer to the memory of a relatively larger computer; for example, to transfer data from a microcomputer to a mainframe computer. [610.5]

up time. The period of time during which a system or component is operational and in service; that is, the sum of busy time and idle time. *Contrast with:* **down time.** *See also:* **busy time; idle time; mean time between failures; setup time.** [610.12]

upward compatible. Pertaining to hardware or software that is compatible with a later or more complex version of itself; for example, a program that handles files created by a later version of itself. *Contrast with:* **downward compatible.** [610.12]

upward compression. In software design, a form of demodularization in which a subordinate module is copied in-line into the body of a superordinate module. *Contrast with:* **lateral compression; downward compression.** [610.12]

usability. The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component. [610.12]

USASCII.* Acronym for **USA Standard Code for Information Interchange.** *See:* **American National Standard Code for Information Interchange (ASCII).** [610.1]
*Deprecated.

USA Standard Code for Information Interchange (USASCII). *See:* **American National Standard Code for Information Interchange (ASCII).** [610.1]
*Deprecated.

user. One who uses the services of a computer system. *Syn:* **end user.** [610.2]

user command. A command that may be issued by the user. For example, "sort" or "print." [610.2]

user-definable key. A key on a computer keyboard that initiates operations or functions that have been defined by the user. *Syn:* **user-programmable key.** [610.2]

user-defined data type. A non-standard data type determined to meet the needs of a particular user or to solve a particular problem. [610.5]

user documentation. Documentation describing the way in which a system or component is to be used to obtain desired results. *See also:* **data input sheet; user manual.** [610.12]

user-driven computing. *Syn:* **end user computing.** [610.2]

user friendly. Pertaining to a computer system, device, program, or document designed with ease of use as a primary objective. *Syn:* **user oriented.** [610.2, 610.12]

user group. An organization of users of a particular class of computer systems, designed to allow the users to share knowledge about and programs for those systems and to formulate feedback for the systems' manufacturers. *Syn:* **user's group.** [610.2]

user guide. *See:* **user manual.** [610.12]

user hotline. Telephone access to a specialist who provides users with answers to questions concerning some product, system, or application. [610.2]

user interaction. Communication between a computer system and a user in which each user entry causes a response from the system. [610.2]

user interface. An interface that enables information to be passed between a human user and hardware or software components of a computer system. [610.12]

user manual. A document that presents the information necessary to employ a system or component to obtain desired results. Typically described are system or component capabilities, limitations, options, permitted inputs, expected outputs, possible error messages, and special instructions. *Note:* A user manual is distinguished from an operator manual when a distinction is made between those who operate a computer system (mounting tapes, etc.) and those who use the system for its intended purpose. *Syn:* **user**

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052952 7

guide. *See also:* **data input sheet; diagnostic manual; installation manual; operator manual; programmer manual; support manual; user manual.** [610.12]

user oriented. *See:* **user friendly.** [610.2]

user program. A computer program written specifically for or by a particular user. [610.2, 610.5a]

user-programmable key. *See:* **user-definable key.** [610.2]

user's group. *See:* **user group.** [610.2]

user state. *See:* **problem state.** [610.12]

user working area. A work area used by a database management system to load and unload data in response to a call by some application program for data. *Syn:* **workspace.** [610.5]

utility. A software tool designed to perform some frequently used support function. For example, a program to copy magnetic tapes. [610.12]

utilization. In computer performance evaluation, a ratio representing the amount of time a system or component is busy divided by the time it is available. *See also:* **busy time; idle time; up time.** [610.12]

V&V. Acronym for **verification and validation.** [610.12]

validation. The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. *Contrast with:* **verification.** [610.12]

validity check. A consistency check that is based upon known limits relating to particular data. For example, a month may not be numbered greater than 12, and week cannot have more than 168 hours. [610.5]

value. *See:* **data value.** [610.5]

value trace. *See:* **variable trace.** [610.12]

variable. A quantity or data item whose value can change; for example, the variable Current_time. *Contrast with:* **constant.** *See also:* **dependent variable; global variable; independent variable; local variable; state variable.** [610.3a, 610.12]

variable format (V format). A file organization in which logical records are of variable length. *Contrast with:* **fixed format.** [610.5]

variable length. Pertaining to a record or field that does not have a constant length, but whose length depends on the length of the specific data contained in it. *Contrast with:* **fixed-length.** *See also:* **variable format.** [610.5]

variable length field. A field whose length may vary according to data stored. *Contrast with:* **fixed length field.** *See also:* **variable format.** [610.5]

variable name data element. A data element whose name can vary depending upon the particular data item represented; for example, a data element named "Population of *X* in *Y*," where *X* takes on the name of a city and *Y* represents a given year. [610.5]

variable point. Pertaining to a numeration system in which the position of the radix point is indicated by a special character at that position. *Contrast with:* **fixed point; floating point.** [610.1, 610.5]

variable trace. A record of the name and values of variables accessed or changed during the execution of a computer program. *Syn:* **data-flow trace; data trace; value trace.** *See also:* **execution trace; retrospective trace; subroutine trace; symbolic trace.** [610.12]

variant. In fault tolerance, a version of a program resulting from the application of software diversity. [610.12]

VDD. Acronym for **version description document.** [610.12]

vector. A quantity represented by an ordered set of numbers; for example, a one-dimensional array. *Contrast with:* **scalar.** *See also:* **column vector; row vector.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

Veitch chart. *See:* **Veitch diagram.** [610.1]

Veitch diagram. A variation of the Karnaugh map in which the rows and columns are headed with combinations of the variables in a straight binary sequence. *Syn:* **Veitch chart; Veitch-Karnaugh diagram.** [610.1]

Veitch-Karnaugh diagram. *See:* **Veitch diagram.** [610.1]

Venn diagram. A diagram in which sets are represented by closed regions. [610.1]

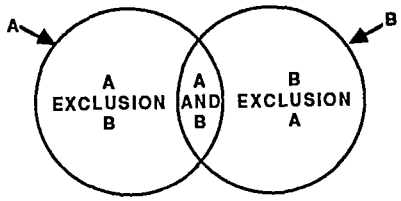


Fig 75
Example of a Venn Diagram

verbal-descriptive model. *See:* **narrative model.** [610.3]

verification. (1) The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. *Contrast with:* **validation.**[610.12] (2) Formal proof of program correctness. *See:* **proof of correctness.** [610.12]

verification and validation (V&V). The process of determining whether the requirements for a system or component are complete and correct, the products of each development phase fulfill the requirements or conditions imposed by the previous phase, and the final system or component complies with specified requirements. *See also:* **independent verification and validation.** [610.12]

version. (1) An initial release or re-release of a computer software configuration item, associated with a complete compilation or recompilation of the computer software configuration item. [610.12] (2) An initial release or complete re-release of a document, as opposed to a revision resulting from issuing change pages to a previous release. [610.12] *See also:* **configuration control; version description document.** [610.12]

version description document (VDD). A document that accompanies and identifies a given version of a system or component. Typical contents include an inventory of system or component parts, identification of changes incorporated into this version, and installation and operating information unique to the version described. [610.12]

vertical justification. In text formatting, justification of text by adding small increments of vertical space between paragraphs and lines to create a well-spaced output page or a series of pages with equal top and bottom margins. [610.2]

vertical microinstruction. A microinstruction that specifies one of a sequence of operations needed to carry out a machine language instruction. *Note:* Vertical microinstructions are relatively short, 12 to 24 bits, and are called "vertical" because a sequence of such instruction, normally listed vertically on a page, are required to carry out a single machine language instruction. *Contrast with:* **diagonal microinstruction; horizontal microinstruction.** [610.12]

vertical tabulation character (VT). A format effector character that causes the print or display position to move to the corresponding position on the next of a series of predetermined lines. [610.5]

V format. *See*: **variable format.** [610.5]

vicenary. (1) Pertaining to a selection in which there are 20 possible outcomes. [610.1] (2) Pertaining to the numeration system with a radix of 20. [610.1]

video conferencing. A form of teleconferencing that uses television to allow participants to see one another. *See also:* **computer conferencing.** [610.2]

videotex. A telecommunication system that allows users to interact with a computer by using a specially equipped television set and a keyboard to access remote data banks and to obtain consumer services such as

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

IEEE TITLE*COMP ** 4805702 0052954 0

electronic mail, teleordering, and bank services. *See also:* **teletext; viewdata.** [610.2]

view. (1) A subset of a relational database, formed by applying relational operations to the base relations represented. *See also*: **logical database.** [610.5]
(2) A subset of a data model. [610.5]
(3) *See*: **external schema.** [610.5]

view integration. The integration of two or more logical views into a single logical view. *Note*: This is generally done in the normalization stage of database design. [610.5]

viewdata. A form of videotex that allows users to access remote data banks via telephone and cable lines. *Contrast with:* **teletext.**[610.2]

virgin medium. A data medium in or on which data have never been recorded. [610.5]

virtual address. In a virtual storage system, the address assigned to an auxiliary storage location to allow that location to be accessed as though it were part of main storage. *Contrast with:* **real address.** [610.12]

virtual attribute. An attribute that is derived from stored data by means of user-defined operations rather than being stored. [610.5]

virtual field. A field that appears to be but is not physically stored; rather, it is constructed or derived from existing data when its contents are requested by an application program. [610.5]

virtual memory. *See:* **virtual storage.** [610.12]

virtual point picture character. *See*: **radix point character.** [610.5]

virtual record. A record that appears to be but is not physically stored; rather, it is constructed or derived from existing data when its contents are requested by an application program. [610.5]

virtual relation. A relation that is not stored in a database in the form in which the user sees it, but is instead derived from base relations using user-defined operations. [610.5]

virtual sequential access method (VSAM). An access method for direct or sequential access to data records on storage devices in which auxiliary storage can be addressed as though it were part of main storage. Pages of data are transferred as needed between auxiliary and main storage. *See also*: **basic sequential access method; indexed sequential access method; queued sequential access method.** [610.5]

virtual storage. A storage allocation technique in which auxiliary storage can be addressed as though it were part of main storage. Portions of a user's program and data are placed in auxiliary storage, and the operating system automatically swaps them in and out of main storage as needed. *Syn:* **multilevel storage, virtual memory.** *Contrast with:* **real storage.** *See also:* **virtual address; paging (2).** [610.12]

virtual time. *See:* **simulated time.** [610.3]

visit. To access the node of a tree during a traversal. [610.5]

voice processing. Information processing in which the human voice is the data input. *See also:* **office automation.** [610.2]

volume. (1) A portion of data that, together with it's data carrier, can be handled as a unit. *Syn*: **physical volume.** [610.5]
(2) A data carrier that is mounted and demounted as a unit; for example a disk pack or a reel of magnetic tape. [610.5]

volume label. *See*: **end-of-volume label.** [610.5]

VSAM. Acronym for **virtual sequential access method.** [610.5]

VT. Acronym for **vertical tabulation character.** [610.5]

WADEX. Acronym for **word and author index.** [610.2]

waiver. A written authorization to accept a configuration item or other designated item which, during production or after having been submitted for inspection, is found to depart from specified requirements, but is

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

nevertheless considered suitable for use as is or after rework by an approved method. *See also:* **configuration control.** *Contrast with:* **deviation; engineering change.** [610.12]

walk-through. A static analysis technique in which a designer or programmer leads members of the development team and other interested parties through a segment of documentation or code, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems. [610.12]

war game. A simulation game in which participants seek to achieve a specified military objective given preestablished resources and constraints; for example, a simulation in which participants make battlefield decisions and a computer determines the results of those decisions. *See also:* **management game.** [610.3]

waterfall model. A model of the software development process in which the constituent activities, typically a concept phase, requirements phase, design phase, implementation phase, test phase, and installation and checkout phase, are performed in that order, possibly with overlap but with little or no iteration. *Contrast with:* **incremental development; rapid prototyping; spiral model.** [610.12]

wearout-failure period. The period in the life cycle of a system or component during which hardware failures occur at an increasing rate due to deterioration. *Contrast with:* **constant-failure period; early-failure period.** *See also:* **bathtub curve.** [610.12]

weekend processing. The operations required to complete a weekly cycle. [610.2]

weekly cycle. One complete execution of a data processing function that must be performed once a week. For example, a weekly payroll system. *See also:* **annual cycle; daily cycle; monthly cycle; quarterly cycle.** [610.2]

weight. (1) In positional representation of numbers, the value of a given digit position. *Syn:* **significance.** [610.1]

(2) For a given node in a tree, the number of terminal nodes in the subtree for that node. [610.5]

weight-balanced tree. A binary tree in which the ratio of the weight of the left subtree to the weight of the right subtree is between the square root of two plus and minus one. *Contrast with:* **height-balanced tree.** [610.5]

WHILE. A single-entry, single-exit loop in which the loop control is executed before the loop body. *Syn:* **pre-tested iteration.** *Contrast with:* **closed loop; UNTIL.** *See also:* **leading decision.** [610.12]

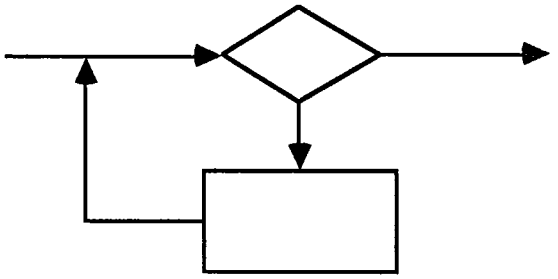


Fig 76
WHILE Construct

white box. *See:* **glass box.** [610.12]

white box model. *See:* **glass box model.** [610.3]

white space expansion. *Syn:* **kerning.** [610.2]

white space reduction. *Syn:* **kerning.** [610.2]

white-box testing. *See:* **structural testing.** [610.12]

widow prevention. The ability of a text formatter to avoid placing a title or the first one or two lines of a paragraph at the end of a page. *See also:* **orphan prevention.** [610.2]

word. (1) A sequence of bits or characters that is stored, addressed, transmitted, and operated on as a unit within a given computer. *Syn:* **computer word; machine word; fullword.** *See also:* **byte.** [610.1, 610.5, 610.12]
(2) An element of computer storage that can hold a sequence of bits or characters as in (1). [610.5, 610.12]
(3) A sequence of bits or characters that has meaning and is considered an entity in

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

some language; for example, a reserved word in a computer language. [610.12] *See also:* **bit; byte.** [610.12]

word and author index (WADEX). A variation of a keyword out of context (KWOC) index in which author and keyword entries are combined and presented in a KWOC format. *Contrast with:* **author and keyword in context index.** [610.2]

word index. An automatic index containing an alphabetical list of the words found in a given text and indicating the number of times each word occurs in the text and each word's position in the text. [610.2]

word processing (WP). The use of computers to enter, view, edit, store, retrieve, manipulate, organize, transmit, and print textual material. A word processor system typically includes text editing and text formatting. *See also:* **clustered word processing; dedicated word processing; office automation; shared-logic word processing; shared-resource word processing; stand-alone word processing; word processor.** *Syn:* **text processing.** [610.2]

word processing output microfilm (WPOM). Microimages produced by a word processor. [610.2]

word processor (WP). (1) A computer capable of performing word processing functions.[610.2] (2) A computer program capable of performing word processing functions. *See also:* **text editor; text formatter.** [610.2]

word wrap. The ability of a word processing system to divide text into lines that fit into the horizontal space available on a display device without leaving broken words or requiring explicit carriage returns. *Syn:* **wraparound.** [610.2]

work file. (1) A file used to provide storage space for data that is needed only during the duration of a particular event, such as the execution of a computer program. [610.5] (2) In sorting, an intermediate file used for temporary storage of data between phases of the sort. [610.5]

working area. *See:* **working space.** [610.12]

working set. In the paging method of storage allocation, the set of pages that are most likely to be resident in main storage at any given point of a program's execution.[610.12]

working space. That portion of main storage that is assigned to a computer program for temporary storage of data. *Syn:* **working area, working storage.** [610.12]

working storage. *See:* **working space.** [610.12]

workload. The mix of tasks typically run on a given computer system. Major characteristics include input/output requirements, amount and kinds of computation, and computer resources required. *See also:* **workload model.** [610.12]

workload model. A model used in computer performance evaluation, depicting resource utilization and performance measures for anticipated or actual workloads in a computer system. *See also:* **system model.** [610.12]

workspace. *See*: **user working area.** [610.5]

work station. A device used to perform tasks such as data processing and word processing. [610.2]

WP. (1) Acronym for **word processing.** [610.2] (2) Abbreviation for **word processor.** [610.2]

WPOM. Acronym for **word processing output microfilm.** [610.2]

wraparound. *See:* **word wrap.** [610.2]

write. To record data in a storage device or on a data medium. *Contrast with*: **read.** *See also*: **delete; store; update.** [610.5, 610.12]

write access. A type of access to data in which data may be written. *See also*: **delete access; read-only access; read/write access; update access.** *Syn*: **write-only access.** [610.5]

write-only access. *See*: **write access.** [610.5]

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

X-datum line. An imaginary line along the top edge of a punch card, used as a reference edge for mark sensing or scanning. [610.2]

XNOR. Acronym for **exclusive NOR.** [610.1]

XOR. Acronym for **exclusive OR.** [610.1]

Y-datum line. An imaginary line along the right edge of a punch card, used as a reference edge for mark sensing or scanning. [610.2]

year-end processing. The operations required to complete an annual cycle. [610.2]

yearly cycle. *Syn:* **annual cycle.** [610.2]

yoked variable. One of two or more variables that are dependent on each other in such a manner that a change in one automatically causes a change in the others. [610.3]

zero-address instruction. A computer instruction that contains no address fields. *Contrast with:* **one-address instruction; two-address instruction; three-address instruction; four-address instruction.** [610.12]

zero complement. *See:* **radix complement.** [610.1]

zero compression. *See:* **zero suppression.** [610.1]

zero elimination. *See:* **zero suppression.** [610.1]

zero fill. To fill the digit positions of a storage medium with the representation of the character zero. **Syn: zeroize.** [610.1, 610.5]

zero proof. A method of checking computations by adding positive and negative values so that if all computations are accurate the total will be zero. [610.1]

zero suppression. The elimination of zeros that have no significance or use, such as zeros to the left of the integral part of a numeral or zeros to the right of the fractional part. *Syn:* **zero compression; zero elimination.** [610.1]

zero-suppression character. A character within a picture specification that represents a decimal digit in which a blank character is used in place of a zero. *Note:* Z, Y, and * are commonly used. [610.5]

zeroize. *See:* **zero fill.** [610.1, 610.5]

zoned decimal data. Integer data in which each decimal digit occupies one byte, the first four bits of which is called the zone portion and the second four bits, the data portion. The zone portion of the lowest-order byte contains the sign of the integer (hexadecimal A, C, E, or F for positive; B or D for negative); otherwise the zone portion contains the binary value 1111. *Syn:* **unpacked decimal data.** [610.5]

decimal	75_{10}
zoned decimal	$0000\ 0111\ 1111\ 0101_2$
decimal	-91_{10}
zoned decimal	$0000\ 1001\ 1011\ 0001_2$

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services

With the Computer field constantly expanding, new terms are being generated and new meanings are being adopted for existing terms at a rapid rate. Computer Professionals need a reliable and current source for standardized definitions to keep up with the state-of-the-art technology being developed in the Computer Industry. IEEE's Standard Glossaries meet this need.

Now, Computer Professionals can have access to all these definitions in one volume — saving time and expense. The *IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries* is a collection of terms from the six existing individual IEEE Standard Glossaries, and documents the latest Computer Industry vocabulary.

■ IEEE Standard Glossary of Computer Terminology (ANSI)

IEEE Std 610.2-1987

Defines Computer Applications terms. Topics covered include automated language processing, automatic indexing, business data processing, character recognition, computer-aided design and manufacturing, office automation, medical applications, control systems, library automation, personal computers, and telecommunications.

■ IEEE Standard Glossary of Modeling and Simulation Terminology (ANSI)

IEEE Std 610.3-1989

Defines Modeling and Simulation terms. Topics covered include general modeling and simulation concepts, types of models and simulations, modeling and simulation variables, game theory, and queueing theory.

■ IEEE Standard Glossary of Image Processing and Pattern Recognition Terminology

IEEE Std 610.4-1990

Defines Image Processing and Pattern Recognition terms.

■ IEEE Standard Glossary of Data Management Terminology

IEEE Std 610.5-1990

Defines Data Management terms. Topics covered include Computer Applications and Software Verification and Validation.

■ IEEE Standard Glossary of Software Engineering Terminology

IEEE Std 610.12-1990

Defines Software Engineering terms. Topics covered include addressing, assembling, compiling, linking, loading, computer performance evaluation, configuration management, software and system testing, software architecture, and software development.

■ IEEE Standard Glossary of Mathematics of Computer Terminology (ANSI)

IEEE Std 1084-1986

Defines Mathematics of Computing terms. Topics covered include Boolean algebra, number systems, computer arithmetic, complementation, arithmetic errors, number conversions, numeric codes, and basic mathematics.

Software Engineering Seminars

IEEE Seminars — Keeping You Competitive through Industry Standards

To advance your professional skills and the competitive stature of your organization, you need to know your standards. What's more, you need to know how to apply them in your environment. IEEE Standards Seminars help you do both. Tailored to meet your particular requirements, IEEE Seminars cover relevant industry standards that add value to your organization.

Throughout the year, IEEE sponsors seminars on Software Engineering. For details, please write to the IEEE Standards Seminar Manager, 445 Hoes Lane, PO Box 1331, Piscataway, NJ 08855-1331 USA. Or in the US and Canada call 1 (800) 678-IEEE. Outside the US and Canada, please call (908) 562-3805.

ISBN 1-55937-079-3

COPYRIGHT Institute of Electrical and Electronics Engineers, Inc
Licensed by Information Handling Services