

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.*

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.

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
Freny Katki
Dr. John Lane

Louise McMonegal
Bennett Meyer
Dr. Hugh Porteous
Jane Radatz

Dr. Fredrick Springsteel
Paul Wilson
Mary Yee

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. Shear
Leonard W. Seagren
Wayne Smith
Elias Tonia
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

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 Elmore
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 Huckleby
Mike Humphrey
John Hyde
James Isaak
David James
Hal Jespersen
Richard Kalish
Matt Kaltenbach
Hans Karlsson
Freni Katki
Guy Kelley
Kim Kirkpatrick
Bob Knighten
Stanley Krolkoski
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
Softtran
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

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

Balloting Committee—610.2-1987, Computer Applications

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

Balloting Committee—610.3-1989, Modeling and Simulation

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



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

David A. Ackley
Tom Armbruster
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. Earls
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

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

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

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 <i>Students</i>	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

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 <i>Students</i> 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 <i>Students</i> 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

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; multi-word 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.

[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]

¹Numbers 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.

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]

- 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]

- (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 (ad hoc)**. 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]

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]

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 $a + bx + cx^2$ 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]

- 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

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]

PARTS INVENTORY

PART NO.	NAME	INVENTORY AMOUNT	SUPPLIER NO.
----------	------	------------------	--------------

SUPPLIER

SUPPLIER NO.	NAME	ADDRESS	NO. ORDERED
--------------	------	---------	-------------

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]

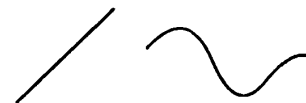


Fig 3
Illustrations of Arc (2)

- 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]

- 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

- 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

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

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]

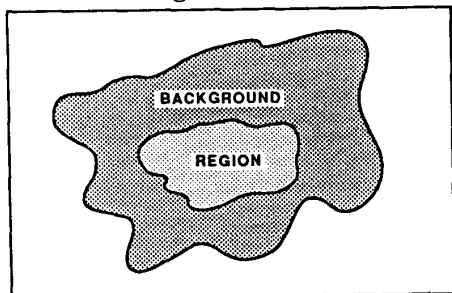


Fig 4
Illustration of Background

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

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

- 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]

- 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-**

- 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

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]

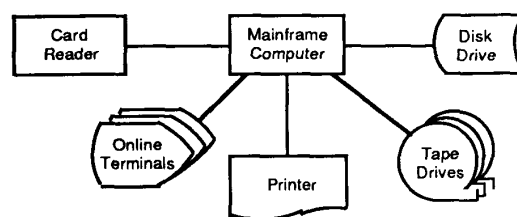


Fig 5
Block Diagram

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]

- 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]

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]

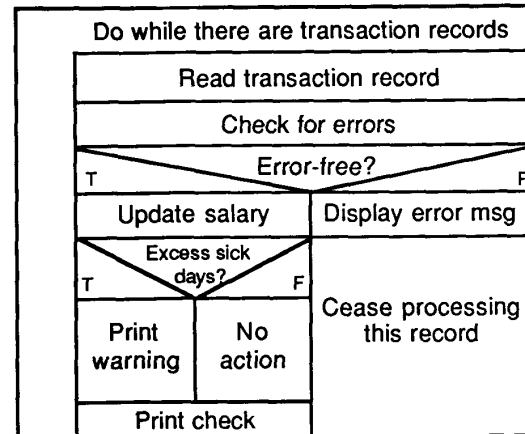


Fig 6
Box Diagram

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]

- (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, program-mable 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 $\{k_1, k_2, \dots, k_m\}$ in ascending order, where $m \leq n - 1$. 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 nodes contain "values less than 10," "11-19," "20-44," "45-59," and "values greater than 60," respec-

tively. *Note:* The height balance of a B-tree is zero. *Syn:* **B-tree index.** *See also:* **B'-tree; B*-tree.** [610.5]

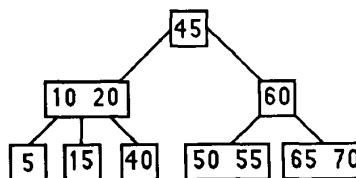


Fig 7
B-Tree of Order 3

(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]

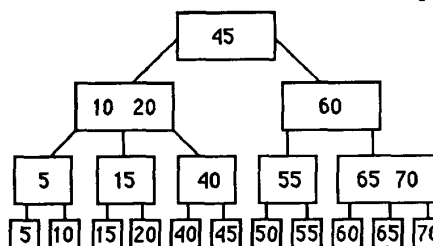


Fig 8
B'-Tree of Order 3

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]

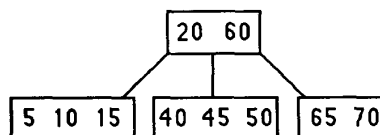


Fig 9
B*-Tree of Order 3

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]

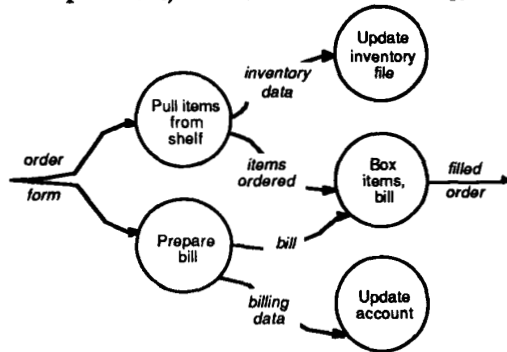


Fig 10
Bubble Chart

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

- 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: goto.* [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

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]

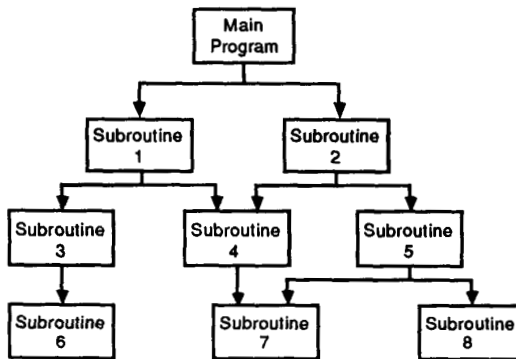


Fig 11
Call Graph

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

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

Augend	289594
Addend	<u>320607</u>
First Partial Sum	509191
First Partial Carry	<u>101010</u>
Second Partial Sum	600101
Second Partial Carry	<u>010100</u>
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]

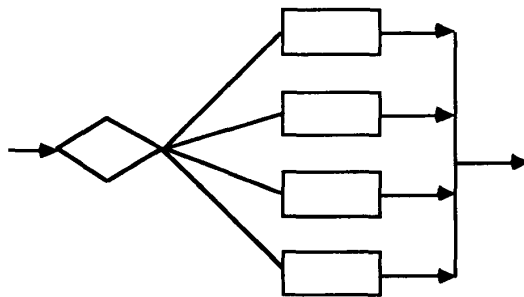


Fig 13
Case Construct

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]

- 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

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

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 <i>x</i>	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]

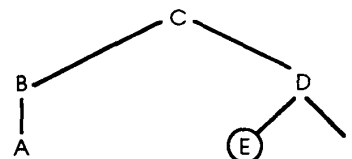


Fig 14
E is a Child Node of Node D

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]

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.**

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]

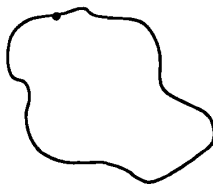


Fig 15
Illustration of Closed Curve

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

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 break-**

point. [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-

- 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

- 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

- 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; meta-compiler.* [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]

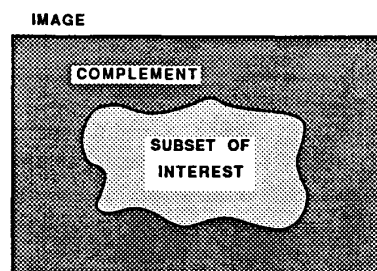


Fig 16
Illustration of Complement

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]

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]

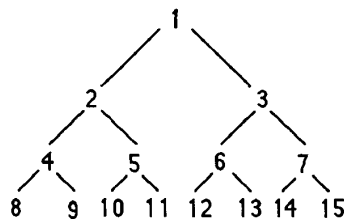


Fig 17
Complete Binary Tree

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]

- 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]

- 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 (CBI).** 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]

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."

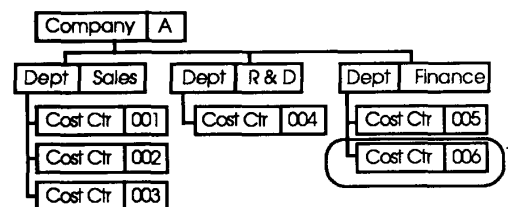


Fig 18
Concatenated Key

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]

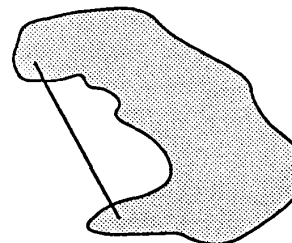


Fig 19
Illustration of Concave

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]

(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 ac-

counting 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]