IEEE Standard Glossary of Computer Languages

Sponsor

Standards Coordinating Committee of the IEEE Computer Society

Approved March 18, 1993

IEEE Standards Board

Abstract: This glossary defines terms that pertain to computer languages. It includes types of computer languages, such as page description languages, and names of computer languages. The languages included are those that are standard languages, languages of historical significance, and those in wide usage at the time this standard was developed.
Keywords: computer glossary, computer languages, computer terminology

This standard is one of a number of information technology dictionaries developed by standards organizations accredited by the American National Standards Institute. These dictionaries are developed under the sponsorship of voluntary standards organizations, using a consensus-based process.

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

ISBN 1-55937-296-6

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 that 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 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
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Introduction

(This introduction is not a part of IEEE Std 610.13-1993, IEEE Standard Glossary of Computer Languages)

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 (the 610 computer glossary series of documents) 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 completed dictionary will contain terms from each of the following areas: computer hardware, software engineering, mathematics of computing, theory of computation, computer applications, artificial intelligence, data management, image processing and pattern recognition, modeling and simulation, computer graphics, computer networking, computer languages, and computer security and privacy. This glossary contains the terms related to computer languages.

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, new, revised, or composite definitions have been developed.

At the time this glossary was approved, the following people formed the steering committee of the computer dictionary working group:

Anne K. Geraci, *Leader, Computer Languages Subgroup*

Other subgroup leaders:

Jack Aldridge
Freny A. Katki
John B. Lane
Louise McMonegal
Hugh L. Porteous
Jane Radatz
Paul C. Wilson
Mary Yee

Other working group members who contributed to this glossary were as follows:

H. Ronald Berlack
J. David Bezek
James H. Bradley
Michael Jesse Choneles
Joel J. Forman
John A. Goetz
Ernesto Guerrieri
David A. Gustafson
Michael L. Morgillo
Gerry R. Neidhart (Zimmerman)
R. Waldo Roth
Charles E. Russell
Max J. Schindler
Paul E. Schmid, III
Wayne Smith
Andrew H. Weigel

The following persons were on the balloting committee:

Jim Barbera
Ralph Barker
Mordechai Ben-Menachem
H. Ronald Berlack
William J. Boll
Kathleen L. Briggs
Fletcher Buckley
Susan Marie Burgess
John Chihorek
Christopher Cooke
Paul I. Davis
L. G. Egan
Philip H. Enslow
John W. Fendrich
Julian Forster
Eugene P. Friedman
Nancy George
Anne K. Geraci
Yair Gershkvitch
Shirley Gloss-Soler
David V. James
Hal Jespersen
Russell E. Jones
Freny A. Katki
Peter Klopfenstein
Thomas M. Kurihara
Lak Ming Lam
John B. Lane
John Lewis
F. C. Lim
Ben Livson
Dieter Look
Alfred Lowenstein
Joseph Maayan
Harold Mains
Roger Martin
Ivano Mazza
Mike McAndrew
Ben Mooy
Gerry R. Neidhart (Zimmerman)
Dennis E. Nickle
Donald J. Pfeiffer
Peter T. Poon
Jane Radatz
Thad L. D. Regulinski
Arthur S. Robinson
Sergiu Samuel
Julio Gonzalez Sanz
Paul E. Schmid, III
Gregory D. Schumacher
Leonard Seagren
David M. Siefert
Robert N. Sulgrove
Donn S. Terry
Richard H. Thayer
Helmut E. Thiess
Glendon R. Trebble
Jon Turino
Mark-Rene Uchida
William M. Walsh
Herbert Weber
Scott A. Whitmire
Andrew Wilson
Weider Yu
Janusz Zalewski

The IEEE Standards Board approved this standard on March 18, 1993, with the following membership:

Wallace S. Read, *Chair* **Donald C. Loughry,** *Vice Chair*
Andrew G. Salem, *Secretary*

Gilles A. Baril
José A. Berrios de la Paz
Clyde R. Camp
Donald C. Fleckenstein
Jay Forster*
David F. Franklin
Ramiro Garcia
Donald N. Heirman
Jim Isaak
Ben C. Johnson
Walter J. Karplus
Lorraine C. Kevra
E. G. "Al" Kiener
Ivor N. Knight
Joseph L. Koepfinger*
D. N. "Jim" Logothetis
Don T. Michael*
Marco W. Migliaro
L. John Rankine
Arthur K. Reilly
Ronald H. Reimer
Gary S. Robinson
Leonard L. Tripp
Donald W. Zipse

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
James Beall
Richard B. Engelman
David E. Soffrin
Stanley I. Warshaw

Christopher J. Booth
IEEE Standards Project Editor

Contents

THIS PAGE WAS
BLANK IN THE ORIGINAL

IEEE Standard Glossary of Computer Languages

1. Overview

1.1 Scope

This glossary defines terms that pertain to computer languages. It includes types of computer languages, such as "page description language," and names of computer languages, such as "BASIC."

In order to qualify for inclusion, a computer language had to be at least one of the following:

a) A standard language, documented and standardized by a nationally or internationally recognized standards organization.
b) A language of historical significance in the evolution of computers and computer languages.
c) A language that is considered by the members of the working group to be in wide usage at this time.

Some terms are the names of commercial products. Many times such products are developed by a particular group, then the product name is accepted as a generic description of all products performing similar functions. For this reason, this glossary does not, in most cases, attempt to identify the owner or developer of a product. The reader should exercise caution in drawing inferences concerning ownership, trademarks, and copyright of names.

Many of the languages defined in this glossary have been standardized by various national and international organizations. Since identification of such languages was not identified as an objective of this document, it does not attempt to identify which languages are considered "standard languages."

The inclusion of a language in this glossary is intended to associate that language with the computer industry and to define and identify its commonly accepted usage. This inclusion does not imply that a given language is the only language that can perform a stated function, nor does it imply a preference for the included language over one that has not been included. If the reader knows of additional languages that should be included in this glossary, it is suggested that such comments be addressed to the IEEE Standards Board, as instructed in the front of this document.

Some technical terms that appear in the definitions are defined in other glossaries in the 610 series and are not included as entries here.

1.2 Glossary structure

Entries in the glossary are arranged alphabetically. An entry may consist of a single word, such as "ALGOL" or "preprocessor," or a phrase, such as "fifth generation language." Phrases are given in their natural order, as in "job control language," rather than in reversed order, "language, job control." Blanks and numerals precede all other characters in alphabetizing. Hyphens and slashes are treated as blanks.

The names of many computer languages are acronyms or abbreviations for phrases that are rarely used, such as "formula translator" for FORTRAN. These phrases are not listed as separate entries, but are noted in the primary entry for the language name. Also, a number of the names are trademarks. However, identification of these has not been made in the glossary.

If a term has more than one definition, the definitions are numbered. The order of the definitions does not imply preference or frequency of use. In most cases, noun definitions are given first, followed by verb and adjective definitions as applicable. Examples and notes have been added to clarify selected definitions.

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

a) *Contrast with:* refers to a term with an opposite or substantially different meaning.
b) *Syn:* refers to a synonymous term.
c) *See also:* refers to a related term.
d) *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.

2. References

In those cases in which a definition is directly quoted from an existing dictionary or glossary, the following references apply:

[610.3] IEEE Std 610.3-1989, IEEE Standard Glossary of Modeling and Simulation Terminology (ANSI).[1]

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

3. Terms and definitions

3.1 1GL. Acronym for **first generation language.** *See:* **machine language.**

3.2 2GL. Acronym for **second generation language.** *See:* **assembly language.**

3.3 3 GL. Acronym for **third generation language.** *See:* **high-order language.**

3.4 4GL. Acronym for **fourth generation language.**

3.5 5GL. Acronym for **fifth generation language.**

3.6 A-0. A programming language developed in 1953 for UNIVAC computers; uses three-address code instructions for solving mathematical problems. *Note:* Developed by Grace Hopper, A-0 was the first computer language for which a compiler was developed.

3.7 ABASIC. A dialect of the BASIC programming language.

3.8 ACSL (Advanced Continuous Simulation Language). A simulation language used for continuous simulation applications.

3.9 ACTOR. An object-oriented language designed to facilitate development of SAA-compliant systems.

3.10 Ada. A programming language designed, developed, and primarily used by the United States Department of Defense. The original design of Ada was based on Pascal, with more complex features such as private data types, synchronized rendezvous for multi-tasking environments, and exception handlers. *Note:* Named after Ada Lovelace, an early pioneer in computing. *See also:* **block-structured language; extensible language; HAL; high order language.**

3.11 ADSIM (Applied Dynamics International Simulation Language). A simulation language designed for use in dynamic simulation applications.

3.12 algebraic language. A programming language that permits the construction of statements resembling algebraic expressions, such as $Y = X + 5$. For example, NOMAD or FORTRAN. *See also:* **algorithmic language; functional language.**

3.13 ALGOL (ALGOrithmic Language or ALGebraic Oriented Language). A high-order programming language suitable for expressing solutions to problems requiring numeric computations, algorithms, or mathematical formulas; its many elegant features and formal syntactic definition have inspired much research in programming language theory. *Note:* Jointly developed by the United States and European communities, ALGOL 60 was the first language standard to be adopted as an ISO standard. As of this writing, ALGOL 68 is the dialect accepted as the latest standard language. *See also:* **BALM; block-structured language; EULER; extensible language; GLYPNIR; GTL; IAL; LCC.**

[1]IEEE publications are available from the Institute of Electrical and Electronics Engineers, Service Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

3.14 ALGOL 58. A dialect of ALGOL; developed as an IEEE standard language in 1958. *See also:* **JOVIAL.**

3.15 ALGOL 60. A dialect of ALGOL that was the first version to be adopted as an ISO language standard for ALGOL. *See also:* **EL1; LEAP; MP; SIMULA.**

3.16 ALGOL 68. A dialect of ALGOL characterized by being the first instance of a complete formal definition language.

3.17 algorithmic language. A programming language designed for expressing algorithms; for example, ALGOL. *See also:* **algebraic language; functional language.**

3.18 ALPHA. An extension to PL/1 providing BNF (backus naur form) parsing capabilities. *Note:* Semantic routines are defined in PL/1 and invoked during the parse.

3.19 ANSI C. A standardized version of C established by ANSI.

3.20 APL (A Programming Language). An interactive programming language with a concise syntax that is well-suited for solving mathematical problems requiring intricate vector or matrix manipulations. *Note 1:* Requires a special keyboard configuration due to its extended character set. *Note 2:* Standardized by ISO/IEC.

3.21 application-oriented language. A programming 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; simulation language; specification language.** [610.12][2]

3.22 APSE (Automatic Programming and Scaling of Equations). A programming language similar to FORTRAN, characterized by its ability to describe equation-oriented specifications used in continuous simulation models.

[2]The numbers in brackets refer to the source document in the 610 series of computer glossary standards from which the definition was taken; a lower-case "a" after the number indicates that an editorial change was made in the definition for inclusion in this document.

3.23 APT (Automatically Programmed Tools). A problem-oriented programming language used for programming numerically controlled machine tools.

3.24 assembler language. *See:* **assembly language.**

3.25 assembly language. A symbolic 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. *See also:* **META 5.** *Syn:* **assembler language; low level language; second generation language.** *Contrast with:* **fifth generation language; fourth generation language; high-order language; machine language.**

3.26 ATLAS (Abbreviated Test Language for All Systems). A test language used by test engineers in controlling automatic test equipment.

3.27 ATOLL (Acceptance Test or Launch Language). A test language used to test applications on the Apollo launch vehicle.

3.28 audit command language (ACL). A high-order programming language used widely in audit applications.

3.29 AUDYSIM (Autodynamics simulation language). A simulation language used in dynamic simulation applications.

3.30 authoring language. An application-oriented programming language used to develop courseware for computer-assisted instruction.

3.31 Autocoder. An early symbolic programming language developed for programming computers.

3.32 AWK. A computer language designed for file processing applications. *Note:* AWK originated in the UNIX environment and was named after its originators, Aho, Wienberger, and Kernighan.

3.33 B. A procedural language used in non-numerical computations; primarily designed for systems programming. *Note:* B is based on BCPL and is a precursor to C.

3.34 backus naur form (BNF). A recursive metalanguage used to specify or describe the syntax of a language in which each symbol, by itself, represents a set of strings of symbols. *Note:* Developed by John

Backus and Peter Naur, BNF was one of the first formal systems developed to specify languages. *Syn:* **backus normal form.** *See also:* **ALPHA.**

3.35 backus normal form (BNF). *See:* **backus naur form.**

3.36 BALM (Block and List Manipulator). A programming language, based on LISP, but containing an ALGOL-like syntax, data types such as vectors and strings, and the ability to write macroinstructions.

3.37 BASIC (Beginner's All-purpose Symbolic Instruction Code). A general-purpose programming language designed for writing programs in scientific and business applications. *Note 1:* Originally developed on a mainframe computer in 1964 at Dartmouth College, BASIC was later implemented as the first high-order language available for a microcomputer. *Note 2:* Numerous implementations of BASIC have been developed for various computers. Examples include ABASIC, MBASIC, S-BASIC, and ZBASIC. *See also:* **common language.**

3.38 BCPL (Bootstrap Combined Programming Language). A recursive computer language used primarily for compiler writing and systems programming. *See also:* **B; CINEMA.**

3.39 BLISS (Basic Language for Implementation of System Software). A programming language designed for writing systems software such as compilers and operating systems.

3.40 block-structured language. A design language 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, C, PL/1, Pascal, MENTOR, and Modula II. *See also:* **structured programming language.** [610.12]

3.41 BMDP (Biomedical Statistics Package). A computer language used widely in biomedical statistical applications.

3.42 BNF. Acronym for **backus naur form.**

3.43 Bookmaster. A text-formatting language developed by IBM; a superset of DCF and GML that allows for elaborate markup of simple text into complex books, with a large degree of output device independence.

3.44 C. A programming language, standardized by ANSI and ISO, designed for systems programming but also well-suited for general problem solving. Features include concise expressions, well-designed control flow and data structures, and a broad range of operators. *Note:* B is an ancestor of C. *See also:* **ANSI C; block-structured language; C++.**

3.45 C++. A general-purpose programming language based on C, characterized by having facilities for performing object-oriented programming.

3.46 CAL (Conversational Algebraic Language). A general-purpose programming language used in time-sharing environments for solving numerical problems.

3.47 CAMAL (CAMbridge ALgebra system). A programming language used to perform large scale formal algebraic manipulation, particularly in celestial mechanics and general relativity.

3.48 CDL (Computer Design Language). A design language for describing or designing computer architectures at the register level.

3.49 CHILL. A high-order language, standardized by CCITT (Consultative Committee on International Telephone & Telegraph, Geneva), used for communication applications.

3.50 CINEMA. A hardware description language with a compiler written in BCPL; contains normal control statements and also statements providing parallel execution of program statements.

3.51 CLIST. A command language used in the IBM MVS environment.

3.52 CLOS (Common LISP Object System). An object-oriented language based on Common LISP.

3.53 COBOL (COmmon Business-Oriented Language). A high-order programming language standardized by ANSI and ISO, designed for business applications. *See also:* **common language; general-purpose programming language; IDS/1.**

3.54 COBOL 85. A dialect of COBOL; developed as a standard language in 1985, and standardized by IEEE, ISO, and ANSI.

3.55 COGO (COordinate GeOmetry). A problem-oriented programming language used to solve coordinate geometry problems in civil engineering applications.

3.56 COMIT. One of the first languages designed to manipulate text strings; provides pattern matching and substitution capabilities.

3.57 command language. A computer language used to express commands to a computer system and to control their execution. For example, job control language, or REXX. *Syn:* **command-level language.** *See also:* **declarative language; interactive language; rule-based language.**

3.58 command-level language. *See:* **command language.**

3.59 common language. Any programming language that is used widely on a variety of computers; For example, BASIC, C, COBOL, and FORTRAN. *See also:* **general-purpose programming language.**

3.60 Common LISP. A dialect of LISP that is widely accepted as the standard language for LISP. *See also:* **CLOS.**

3.61 compiler specification language. A specification language used to develop compilers. *See also:* **LEX; YACC.**

3.62 computer language. (1) A language designed to enable humans to communicate with computers and computer systems.
(2) A language that is used to control, design, or define a computer or computer program.

3.63 continuous simulation language. A simulation language designed for use in describing continuous simulations.

3.64 Coursewriter. A programming language used to write instructional programs for computer-assisted instruction.

3.65 CSMP III (Continuous System Modeling Program III). A simulation language used to simulate the dynamics of continuous systems that use ordinary differential equations.

3.66 CSS/II (Computer System Simulation II). A simulation language that is based on the concepts used in GPSS, but specialized for use in modeling computer systems.

3.67 CSSL (Continuous Systems Simulation Language). A statement-oriented simulation language used to simulate the dynamics of continuous systems that are describable by ordinary differential equations.

3.68 CYPHERTEXT. A text-formatting language commonly used for typesetting.

3.69 D-TRAN (Decision Table Translator). A computer language developed as a preprocessor that converts decision table constructs into conventional programming language code.

3.70 DARE (Differential Analyzer Replacement). A series of continuous simulation languages for use in batch and on-line applications.

3.71 database description language. A programming language used to provide a programmer with access to a database; the language describes exactly how the data is stored in the database.

3.72 database manipulation language. A computer language used to query, retrieve, insert, delete, or modify items in a database. *See also:* **Datatrieve; dBase; DL/I; Easytrieve; FOCUS; INQUIRE; MODEL 204; NATURAL; RAMIS; SQL.**

3.73 Datatrieve. A database manipulation language used primarily for database applications under Digital's VAX/VMS environment.

3.74 dBASE. A database manipulation language used commonly for database applications in the PC DOS Environment. *Note:* Several version have been published with Roman numeral suffixes, as in dBASE III, dBASE IV.

3.75 DBL. A programming language used in conjunction with DIBOL for developing applications within the UNIX and DOS environments.

3.76 DCDL (Digital Control Design Language). A simulation language for use in designing digital computer systems.

3.77 DCF (Document Composition Facility). A text-formatting language, developed by IBM, for use as an engine for GML to generate formatted documents. *See also:* **SCRIPT.**

3.78 DCL (Digital Command Language). A command language used under Digital's VAX/VMS environments.

3.79 declarative language. (1) A nonprocedural language that permits the user to declare a set of facts and to express queries or problems that use these facts. *See also:* **command language; interactive language; rule-based language.**
(2) A programming language that can be understood without reference to the behavior of any particular computer system.

3.80 design language. A specification language with special constructs and, sometimes, verification protocols, used to develop, analyze, and document a hardware or software design. Types include hardware design language, program design language. *See also:* **CDL; requirements specification language.** [610.12]

3.81 DETOL (Directly Executable Test Oriented Language). A test language used to control a specific type of automatic test equipment.

3.82 dialect. (1) In computer languages, a variation of a particular language. *Syn:* **variation; variant; version.**
(2) A form of a particular language, peculiar to a specific population or group, differing from some standard language in some significant manner. *See also:* **extension; subset.**

3.83 DIBOL (Digital Business Oriented Language). A problem-oriented programming language developed by Digital Equipment Company; used to develop business applications. *See also:* **DBL.**

3.84 DL/I (Data Language I). A database manipulation language used with IMS hierarchical databases.

3.85 DMAD (Diagnostic Machine Aid—Digital). A test language used for functional testing of digital devices; allows device description in terms of registers, signal names, and functional operators, such as logical operators and Boolean operations.

3.86 DSL (Digital Simulation Language). A simulation language that represents blocks, switching functions, and function generators, similar to those available with an analog computer.

3.87 DYNAMO. A programming language for continuous simulations; used to construct large multisector models of economic, industrial, and social systems, as well as other continuous closed-loop information feedback systems. *Note:* DYNAMO was written entirely in FORTRAN.

3.88 Easytrieve. A database manipulation language used for extracting data from data files and databases.

3.89 ECAP II (Electronic Circuit Analysis Program II). A simulation language used for modeling and analyzing electrical networks, allowing synthesis of device models using a function generator.

3.90 ECSS II (Extendible Computer System Simulator II). An extension of SIMSCRIPT providing statements and data structures for simulating computer hardware configurations, software, and work load.

3.91 EIFFEL. An object-oriented programming language.

3.92 EL1. An extensible language that includes most of the concepts of ALGOL 60 and LISP, but with a syntax similar to ALGOL.

3.93 EOL-3 (Expression-Oriented Language 3). A programming language used to manipulate strings of characters.

3.94 ESP (Econometric Software Package). A programming language used for statistical analysis of time series and other data by regression and more sophisticated econometric techniques. Includes data editing, transformation and display, matrix manipulation, and a variety of complex forecasting procedures.

3.95 Estelle. A specification language for telecommunications and distributed systems based on extended state transitions.

3.96 ETC (Extendible Compiler). An extensible language whose extended language is similar to PL/1. Contains provisions for a programmer to code machine-dependent statements to maximize efficiency.

3.97 EULER. An experimental programming language that is a generalization of the formal definition of ALGOL.

3.98 EXEC. A command language used in IBM's VM/CMS environment to carry out command level processing. *Note:* EXEC was superseded by EXEC2 and is superseded by REXX.

3.99 EXEC2. *See:* **EXEC.**

3.100 extensible language. A computer language that can be altered or can alter itself to provide a programmer with additional user-specified functions or capabilities. Examples include Ada, ALGOL, FORTH, and LOGO, because each can be used in a building block fashion to construct increasingly complex functions. *See also:* **EL1; ETC; MADCAP; MP; PPL.**

3.101 extension. A dialect of a particular language that varies from its referenced standard language such that the extension language has all the capabilities of the referenced language plus some additional capabilities. For example, ALPHA is an extension of PL/1. *Contrast with:* **subset.**

3.102 FGRAAL (FORTRAN Extended GRAph Algorithmic Language). An extension of FORTRAN used widely to solve graph problems. *Note:* Includes facilities for manipulating sets and graphs.

3.103 fifth generation language (5GL). A computer language that incorporates the concepts of knowledge-based systems, expert systems, inference engines, and natural language processing. *Contrast with:* **assembly language; fourth generation language; high-order language; machine language.**

[610.12]

3.104 first generation language (1GL). *See:* **machine language.**

3.105 FLAP. A programming language used widely for manipulating formulas and performing symbolic mathematical calculations.

3.106 FLAVORS. An object-oriented language originally developed as an extension of LISP.

3.107 Flow-matic (Flowmatic). The first automatic programming language, developed specifically for the UNIVAC II computer.

3.108 FOCUS. A fourth-generation language used to develop information systems, characterized by its integrated database manipulation language and its ability to be used on a wide range of computer platforms.

3.109 FOIL (File-Oriented Interpretive Language). A computer language, based on FORTRAN, used to provide conversational lesson-writing; used commonly in computer-aided instruction applications.

3.110 FORMAC (FORmula Manipulation Compiler). An extension of PL/1 used to perform symbolic manipulation of mathematical expressions.

3.111 FORMAL (FORmula Manipulation Language). An extension of FORTRAN used to perform formal algebraic manipulations.

3.112 formal language. A language whose rules are explicitly established prior to its use. Examples include programming languages and mathematical languages. *Contrast with:* **natural language.**

[610.12]

3.113 FORTH. A high-order programming language that can be used for a wide range of applications due to its ability to be used as an interpreter, command language, and even an operating system. *Note:* FORTH is not an acronym. *See also:* **extensible language; Polyforth.**

3.114 FORTRAN or Fortran (FORmula TRANslator). A high-order programming language used widely for solving scientific, mathematical and numerical problems. *Note:* At the time that this standard was written, FORTRAN 77 and Fortran 90 were both accepted IEEE language standards. *See also:* **algebraic language; common language; DYNAMO; FGRAAL; FOIL; FORMAL; GASP IV; GSP; HOS-STPL; LRLTRAN; QUIKTRAN; RATFOR; WATFOR.**

3.115 FORTRAN 66. A dialect of FORTRAN; developed as a standard language in 1966.

3.116 FORTRAN 77. A dialect of FORTRAN; developed as a standard language in 1977.

3.117 Fortran 90. A dialect of FORTRAN; developed as a standard language in 1990.

3.118 FORTRAN IV. A dialect of FORTRAN, developed as a standard language in 1962.

3.119 fourth generation language (4GL). A computer language designed to improve the productivity achieved by high-order (third generation) languages and, often, to make computing power available to non-programmers. Features typically include an integrated database management system, query language, report generator, and screen definition facility. Additional features may include a graphics generator, decision support function, financial modeling, spreadsheet capability, and statistical analysis function. *Contrast with:* **assembly language; high-order language; fifth generation language; machine language.** [610.12]

3.120 functional language. A programming language used to express programs as a sequence of functions and function calls. Examples include LISP and C. *See also:* **algebraic language; algorithmic language.** [610.12]

3.121 GAMMA 3. A programming language used for generating matrices and reports in conjunction with a mathematical programming system.

3.122 GASP IV. A simulation language designed to be used within a FORTRAN program to facilitate the representation of discrete, continuous, and combined models.

3.123 general-purpose programming language. A programming language that provides a set of processing capabilities applicable to most information processing problems and that can be used on many kinds of computers. For example, Ada, COBOL, FORTRAN, and PL/1. *See also:* **CAL; common language; JOSEF; Pascal; SIMULA.**

3.124 GIRL. ***(1) (Graphical Information Retrieval Language).*** A programming language used to manipulate information in arbitrary directed-graph structures, including facilities for insertion, retrieval, deletion, and comparison.
(2) (Generalized Information Retrieval Language). A query language used by the United States Defense Nuclear Agency for information retrieval.

3.125 GLYPNIR. A programming language with syntax similar to that of ALGOL, but with facilities to allow the programmer to specify the parallelism of an algorithm.

3.126 GML (Generalized Markup Language). A page description language used to provide simplified tags in DCF for formatting documents. *See also:* **Bookmaster.**

3.127 GPSS (General Purpose Systems Simulation). A problem-oriented language used in performing discrete simulation problems, based on a block diagram approach, where each block represents a physical process and transactions move from one block to another. *See also:* **CSS/II.**

3.128 GSP (General Space Planner). A programming language based on FORTRAN that provides an interactive system for solving space planning problems.

3.129 GTL (Georgia Tech Language). An extension to ALGOL that contains access to LISP and other facilities.

3.130 GW BAsic. A dialect of BASIC, designed for use with microprocessors and microcomputers.

3.131 Gypsy. A specification language used primarily for computer security applications; one of the two specification languages accepted for use by the US National Computer Security Center.

3.132 HAL. (1) A high-order problem-oriented language used in aerospace applications; characterized by its strong orientation toward mathematical computations and built-in vector/matrix arithmetic.
(2) Abbreviation for High-order Ada Language; a computer language similar to Ada.

3.133 hardware design language (HDL). A specification language with special constructs and, sometimes, verification protocols, used to develop, analyze, and document a hardware design. *Contrast with:* **program design language.** *See also:* **CINEMA.**

3.134 high level language. *See:* **high-order language.**

3.135 high-order language (HOL). Any programming language that requires little knowledge of the computer hardware on which a program will run, can be translated into several different machine languages, allows symbolic naming of operations and addresses, provides features designed to facilitate expression of data structures and program logic, and usually results in several machine instructions for each program statement. Examples include Ada, ALGOL, COBOL, FORTRAN, Pascal. *Syn:* **high**

level language; higher order language; third generation language. *Contrast with:* **assembly language; fifth generation language; fourth generation language; machine language.** [610.12]

3.136 higher order language. *See:* **high-order language.**

3.137 HOS-STPL (Hospital Operating System-Structured Programming Language). An application-oriented language with some FORTRAN features, used to manipulate string variables and to support structured programming.

3.138 HPCL (HPPCL, Hewlett-Packard Printer Control Language). A page description language used in many laser printers.

3.139 HPGL (Hewlett-Packard Graphics Language). A page description language used by many laser printers.

3.140 IAL (International Algebraic Language). A forerunner of the ALGOL language.

3.141 ICES (Integrated Civil Engineering System). A general-purpose software system including several programming languages, such as COGO and STRUDL, and subsystems that are designed for use in civil engineering and engineering management.

3.142 Icon. A high-order programming language designed primarily to process non-numerical data, as in the applications such as analyzing natural language, transforming or generating computer programs, and formatting documents.

3.143 IDS/1. An extension to COBOL that permits data to be represented in ring type lists.

3.144 IITRAN. A programming language similar to PL/1; designed for use as an educational tool.

3.145 INA JO. A computer language used for proving or verifying program correctness.

3.146 INQUIRE. A nonprocedural database manipulation language used to access data stored in VSAM databases; characterized by its suitability for storage and retrieval of textual data.

3.147 INTELLECT. A natural language front-end processor for an SQL database manipulation language.

3.148 interactive language. A nonprocedural language in which a program is created as a result of interactive dialogue between the user and the computer system. The system provides questions, forms, and so on, to aid the user in expressing the results to be achieved. *See also:* **command language; declarative language; LCC; PPL; rule-based language.** [610.12]

3.149 IPL (Information Processing Language). A high-order language used for performing list processing.

3.150 IT (Internal Translator). A programming language developed to handle numerical applications, scientific applications, or expressions evaluated from left to right ignoring operator precedence.

3.151 JCL. An acronym for **Job Control Language.** *See:* **job control language**.

3.152 job control language (JCL). A command language used to identify a sequence of jobs, describe their requirements to an operating system, and control their execution. *Note:* Commonly used in batch-oriented environments such as IBM's 370 Computer. [610.12]

3.153 JOSEF. A general-purpose programming language similar to Pascal.

3.154 JOSS (JOHNNIAC Open Shop System). A procedural language used for performing numerical computations and mathematics. *See also:* **LCC.**

3.155 JOVIAL (Jules' Own Version of International Algorithmic Language). A high-order programming language used primarily for solving scientific and control problems. *Note:* Based on ALGOL 58. *See also:* **TINT.**

3.156 L (Bell Laboratories' Low-level Linked List Language). A list processing language that allows programmers to specify list sizes and types.

3.157 language. (1) A systematic means of communicating ideas by the use of conventional signs, sounds, gestures, or marks, and rules for the formation of admissible expressions.
(2) A means of communication, with syntax and semantics, consisting of a set of representations, conventions, and associated rules used to convey information. *See also:* **computer language.** [610.12]

3.158 language-description language. *See:* **meta-language.**

3.159 language standard. A standard that describes the characteristics of a language used to describe a requirements specification, a design, or test data. *See also:* **standard language.** [610.12]

3.160 LaTeX. A text-formatting language based on TeX.

3.161 LCC (Language for Conversational Computing). An interactive programming language combining ALGOL-like syntax with many of the features of JOSS.

3.162 LEAP (Language for the Expression of Associative Procedures). A programming language based on ALGOL 60, containing set-theoretic and associative operations and data types.

3.163 LEX or Lex. A compiler specification language in which the input is (1) a specification of a set of regular expressions and (2) actions to be taken upon recognizing each of these. The output of Lex is a lexical analysis program that can process the specified language. *Note:* Used in writing portions of compilers, as well as in textual pattern matching. *See also:* **YACC.**

3.164 LISP (LISt Processing). A list processing language designed for manipulating symbols and for operating on strings of information known as "lists"; handles recursive and repetitive handling of connected character strings. *Note:* Used widely in artificial intelligence, LISP uses a functional notation derived from lambda calculus that permits programs and data to have the same structure. *See also:* **BALM; Common LISP; EL1; FLAVORS; functional language; GTL; SAIL; SCHEME.**

3.165 list processing language. A programming language designed to manipulate data expressed in the form of lists or character strings. Examples are LISP, LOGO, LPL, PROLOG, and SAM76. *Syn:* **symbolic language; symbol manipulation language.** [610.12]

3.166 logic programming language. A programming language used to express programs in terms of control constructs and a restricted predicate calculus; for example, PARLOG; PROLOG or STRAND.

3.167 LOGO. A high-order list processing language designed for interactive educational applications; characterized by its simple vocabulary and built-in graphics capability, known as "turtle graphics." *See also:* **extensible language.**

3.168 LOOPS. An object-oriented language designed as an expert system shell.

3.169 LOTOS (Language of Temporal Ordering Specifications). A specification language used for telecommunications and distributed systems.

3.170 low level language. *See:* **assembly language.**

3.171 LPL (List Processing Language). An extension of PL/1 used to provide list processing facilities in which the user can define and operate on cells of varying characteristics.

3.172 LRLTRAN (Lawrence Radiation Laboratory TRANslator). A FORTRAN-based language used to perform vector arithmetic, bit and byte manipulation, and pointer manipulation.

3.173 LYRIC (Language for Your Remote Instruction by Computer). An application-oriented language used primarily for computer-assisted instruction.

3.174 machine language. A programming language that is directly executed by the central processing unit (CPU) portion of a computer. *Note:* No further translation, mapping, or decoding is required. *Syn:* **first generation language; machine-oriented language.** *Contrast with:* **assembly language; fifth generation language; fourth generation language; high-order language; symbolic language.**

3.175 machine-oriented language. *See:* **machine language.**

3.176 MACRO. A macro language used in Digital's VAX/VMS environment.

3.177 macro language. A language used to define macros or macroinstructions. *Note:* IEEE Std 610.12-1990 defines terminology relating to macroinstructions.

3.178 MACSYMA. An interactive programming system used to perform formal algebraic manipulation and symbolic mathematics.

3.179 MAD (Michigan Algorithmic Decoder). A programming language used widely for doing numerical computations. *Note:* MAD was designed to permit the development of a very fast compiler. *See also:* **NOMAD.**

3.180 MADCAP. An extensible language used to perform numerical computation and set-theoretic operations, using input and output devices that permit two-dimensional input and output.

3.181 MAPPER (Maintaining, Preparing & Producing Executive Reports). A nonprocedural programming language for UNIVAC computers, designed for novice users.

3.182 MARK IV. A procedural language used for report writing and data manipulation.

3.183 markup language. *See:* **page description language.**

3.184 MARSYAS (MARshall System for Aerospace Simulation. A simulation language used for simulating large physical systems, designed for use by people inexperienced in simulation or programming. Allows equations and FORTRAN subroutines to be written along with the statements describing a block diagram model.

3.185 Mathematica. A programming language designed to manipulate equations symbolically.

3.186 MBASIC. A dialect of the BASIC programming language.

3.187 MENTOR. A block-structured language used widely in computer-aided instruction; characterized by its ability to model a student's knowledge.

3.188 Mesa. An application development language used by Xerox to program Viewpoint applications.

3.189 META 5. (1) A programming language used for symbolic data manipulation and for syntax-directed computing.
(2) An assembly language for CDC computers.

3.190 metalanguage. A language used to specify some or all aspects of itself or of another language; for example, backus naur form. *See also:* **stratified language; unstratified language; VDL.** *Syn:* **language-description language.**

3.191 MIMIC. A problem-oriented programming language for solving engineering problems, particularly those involving differential equations.

3.192 MOBSSL-UAF (Merritt and Miller's Own Block Structured Simulation Language). A simulation language used to model continuous systems using an augmented block structure.

3.193 MODEL. An application-oriented language used widely for simulating digital circuits.

3.194 MODEL 204 (M204). A database manipulation language with English-like syntax.

3.195 MODULA II (Modula 2, or MODUlar LAnguage II). A programming language developed, as an expanded version of Pascal, to support modular design, structured programs, and mathematical calculations. *See also:* **block-structured language.**

3.196 MP. A dialect of ALGOL 60 having extensible language features; used largely as a programming language for system software.

3.197 MPSX (Mathematical Programming System Extended). A programming language used widely for controlling the solution strategy for mathematical programming problems.

3.198 MUMPS (Massachusetts General Hospital Utility Multi-Programming System). An ANSI standard programming system containing its own operating system, command language, and interactive programming language; designed specifically for medical applications and is particularly adaptable to string handling functions and management of hierarchical data.

3.199 NATURAL. A database manipulation language used to access data stored in an ADABAS database.

3.200 natural language. A stratified language whose rules are based on usage rather than being preestablished prior to the language's use. Examples include German and English. *Contrast with:* **formal language.** *See also:* **INTELLECT.**

3.201 NELIAC (Naval Electronics Laboratory International Algorithmic Compiler). A programming language used primarily for solving scientific and real-time control problems.

3.202 NOMAD. A fourth-generation programming language that permits a wide latitude of generality in algebraic expressions. *Note:* Adapted from **MAD.**

3.203 nonprocedural language. A computer language in which the user states what is to be achieved without having to state specific instructions that the computer must execute in a given sequence. *Contrast with:* **procedural language.** [610.12]

3.204 nroff. A text-formatting language.

3.205 Objective C. An object-oriented version of C.

3.206 object-oriented language. A computer language that allows the user to express a program in terms of objects and messages between those objects. Examples include SMALLTALK and LOGO. *See also:* **C++; EIFFEL; FLAVORS; LOOPS.**

3.207 Occam. A general-purpose programming language designed in the early 1980's for use in parallel computer systems.

3.208 OL/2. A programming language designed to allow statement of mathematical problems, with emphasis on arrays and structures that exhibit the parallelism inherent in many algorithms.

3.209 OMNITAB II. A programming language designed for nonprogrammers, to provide data, numerical, and statistical analysis; provides capability for performing calculations and statistical procedures such as regression and matrix inversion.

3.210 OPS5 (Official Production System). A nonprocedural programming language that uses precise rules, in the form of a rule-and-fact set model, to reach solutions to problem descriptions. *Note:* Used in artificial intelligence applications for building expert systems.

3.211 OSCAR (Oregon State Conversational Aid to Research). An interactive programming system used for performing numerical calculations, string manipulations, vector and matrix operations, and complex arithmetic.

3.212 OSSL (Operating Systems Simulation Language). A simulation language used to simulate hardware and software aspects of computer systems.

3.213 PAGE. A text-formatting language that uses two-character instruction codes to control typesetting.

3.214 page description language. A computer language in which commands from a text-formatting language are combined into higher-level instructions that can be used in other documents. Examples include GML, HPGL, Postscript, and TEX. *Syn:* **markup language.**

3.215 PARLOG. A logic programming language used widely for parallel computing, supporting declarative programming.

3.216 PARSEC (PARser and Extensible Compiler). An extensible language using syntax similar to PL/1; PARSEC is derived from PROTEUS and is used as the base language for writing PL/PROPHET.

3.217 Pascal. A general-purpose programming language standardized by IEEE & ASC X3 adapted for use on a variety of computers; characterized by its ability to handle algorithms, various data types, and block-structured. *Note:* Often used in teaching programming concepts, Pascal was named after Blaise Pascal, a French mathematician, and was developed by Niklaus Wirth. *See also:* **block-structured language; high-order language; JOSEF; MODULA II; Rascal.**

3.218 PCL. *See:* **HPCL.**

3.219 PDEL (Partial Differential Equation Language). An application-oriented language used for solving partial differential equations in which the user does not have to program the numerical analysis algorithms. *Note:* Used as a preprocessor to PL/1. *Syn:* **PDELAN.**

3.220 PDELAN (Partial Differential Equation LANguage). *See:* **PDEL.**

3.221 PDL. Abbreviation for **program design language.**

3.222 PDS/MaGen (Problem Descriptor System). A programming language useful in a wide variety of operations research applications, and designed to facilitate the generation of matrices and reports for mathematical programming systems.

3.223 PEARL (Process and Experiment Automation Realtime Language). A general-purpose, high-

order language designed to meet the requirements of real-time programming in process and experiment automation.

3.224 PILOT (Programmed Inquiry, Learning Or Teaching). A programming language designed for writing computer-aided instruction applications; PILOT is simple and well-suited to support an interactive "question and answer" type of system.

3.225 PL/1 (Programming Language/1). A programming language that is suitable for processing numerical, scientific, and business applications and that is standardized by ANSI. *See also:* **ALPHA; block-structured language; FORMAC; IITRAN; LPL; VDL.**

3.226 PL/C. A subset of PL/1 characterized by its enhanced debugging, quick compilation, code optimization and error checking facilities. *Note:* The C stands for Cornell University, where the language was developed.

3.227 PL/DB (Programming Language/Data Base). A dialect of PL/1 designed specifically for processing databases and including normal executable statements for arithmetic, conditional, and loop control and supports hierarchical data structures.

3.228 PL/I. Deprecated for PL/1.

3.229 PL/M. A procedure-oriented programming language derived from PL/1, designed specifically for microcomputers.

3.230 PL/PROPHET. A language similar in style to PL/1, used in conjunction with the PROPHET system in pharmacology research. *See also:* **PARSEC.**

3.231 PLANIT (Programming LANguage for Interactive Teaching). An instructional system consisting of a user language that supports the development of computer programs for preparing, editing, and presenting subject matters suitable for interactive presentations.

3.232 PLANNER. A computer language and reasoning model used commonly in artificial intelligence for proving theorems and for manipulating models in a robot.

3.233 PLI. Deprecated for PL/1.

3.234 Polyforth. A dialect of FORTH.

3.235 Postscript. A page description language used in many laser printers.

3.236 PPL (Polymorphic Programming Language). An interactive, extensible language containing facilities for defining new data types and operators.

3.237 PREP (PRogrammed Electronics Patterns). A programming language for use in designing integrated circuits. *Note:* PREP is conceptually similar to APT, except that it involves description of two-dimensional figures.

3.238 preprocessor. A computer program or routine that carries out some processing step prior to the primary process; for example, a precompiler or other routine that reformats code or data for processing. *See also:* **D-TRAN; PDEL; LYRIC.** [610.12]

3.239 problem-oriented language. A programming language designed for the solution of a given class of problems; for example, list processing languages or simulation languages.

3.240 procedural language. A computer language in which the user states a specific set of instructions that the computer must perform in a given sequence. Examples include BASIC, COBOL, FORTRAN, and Pascal. *Syn:* **procedure-oriented language.** *Contrast with:* **nonprocedural language.**

3.241 procedure-oriented language. *See:* **procedural language.**

3.242 PROFILE. A computer language used to match, score, and retrieve statistical data.

3.243 program design language (PDL). A specification language with special constructs and verification protocols, used to develop, analyze, and document a program design. *Contrast with:* **hardware design language.** *See also:* **pseudo code.**

3.244 programming language. A computer language used to express computer programs. *See also:* **assembly language; common language; general-purpose programming language; high-order language; machine language.** *Contrast with:* **design language; query language; specification language; test language.** [610.12]

3.245 PROLOG (PROgramming in LOGic). A declarative programming language that uses precise

rules, in the form of a rule and fact set model, to reach solutions to problem descriptions. *Note:* Used in artificial intelligence applications for building expert systems.

3.246 PROTEUS. A computer language used in signal processing.

3.247 pseudo code (pseudocode). A combination of programming language constructs and natural language used to express a computer program design. For example:

```
IF the data arrives faster than expected,
  THEN reject every third input.
  ELSE process all data received.
ENDIF
```

See also: **program design language.**

3.248 query language. A database manipulation language used to access information stored in a database. *Contrast with:* **programming language; specification language.**

3.249 QUIKTRAN (Quick FORTRAN). A subset of FORTRAN that is easy to learn due to its efficient debugging facility and interactive interface.

3.250 RAMIS (Rapid Access Management Information System). A nonprocedural database manipulation language that provides data management and decision support facilities.

3.251 Rascal. A dialect of Pascal. *See:* **Pascal.**

3.252 RATFOR (Rational FORTRAN). An extension of FORTRAN that provides free format coding (as opposed to FORTRAN's strict column format), source file inclusion, and block structures.

3.253 REDUCE. A list processing language written in LISP, used primarily for performing symbolic operations and simplification of arrays and matrices.

3.254 report writer. A software tool or programming language used specifically for generating reports. *See:* **RPG.**

3.255 requirements specification language. A specification language with special constructs and, sometimes, verification protocols, used to develop, analyze, and document hardware or software requirements. *See also:* **design language.** [610.12]

3.256 REXX. A command language used primarily in the IBM VM/CMS environment. *Note:* Supersedes EXEC and EXEC2.

3.257 roff. A text-formatting language.

3.258 RPG (Report Generation Language). A problem-oriented language designed for file processing and report creation.

3.259 rule-based language. A nonprocedural language that permits the user to state a set of rules and to express queries or problems that use those rules. *See also:* **command language; declarative language; interactive language.** [610.12]

3.260 S. A programming language used widely in the UNIX environment for data analysis and visualization.

3.261 S-BASIC. A dialect of the BASIC programming language.

3.262 SAIL (Stanford Artificial Intelligence Language). A dialect of LISP that was developed at Stanford's Artificial Intelligence Laboratory.

3.263 SAM76. A list processing language widely used in artificial intelligence due to its unique suitability for interactive and user-directed applications.

3.264 SAS (Statistical Analysis System). A programming language used for statistical analysis, data manipulation, and application development.

3.265 SCHEME. A dialect of LISP.

3.266 SCRIBE. A text-formatting language in which formatting commands are embedded in the text, then processed into a formatted document.

3.267 SCRIPT. A text-formatting language in which formatting commands are embedded in the text, then processed into a formatted document. *Note:* SCRIPT is a forerunner to DCF.

3.268 SDL (Specification and Description Language). A specification language for telecommunications and distributed systems that provides both textual and graphic description techniques.

3.269 second generation language (2GL). *See:* **assembly language.**

3.270 SGML (Standard Generalized Markup Language). A text-formatting language.

3.271 shell. (1) A software interface between the user and the operating system in which the shell interprets commands and communicates them to the operating system of the computer.
(2) Software that allows a kernel program to run under different computing environments.

3.272 SIM++. A programming language used for simulations on distributed computing systems.

3.273 SIMSCRIPT. A high-order language designed for use in performing general-purpose digital simulations. *Note:* Allows for the description of a system in terms of its "attributes," which are properties associated with "entities," which are groups of "sets." *See also:* **ECSS II.**

3.274 SIMULA. A general-purpose programming language based on ALGOL 60, with special features designed to aid the description and simulation of active processes.

3.275 simulation language. An application-oriented programming language used to implement simulations. *See also:* **continuous simulation language.**
[610.3a]

3.276 SLIP (Symmetric List Processing Language). A high-order list processing language using a structure reader, a parser that can traverse a data structure.

3.277 SMALLTALK. A high-order language based on the metaphor of objects sending messages to one another. *See also:* **object-oriented language.**

3.278 SNOBOL (StriNg-Oriented symBOLic Language). A programming language designed for use in string manipulation tasks such as language translation, program compilation and combinatorial problems. *See also:* **SPITBOL.**

3.279 specification language. An application-oriented computer language, often a machine-processible combination of natural and formal language, used to express the requirements, design, behavior, or other characteristics of a system or component. For example, a design language or requirements specification language. *Contrast with:* **compiler specification language; programming language; query language.** *See also:* **design language.**

3.280 SPICE (Simulation Program with Integrated Circuit Emphasis). A simulation language used widely to design electrical circuits.

3.281 SPITBOL (SPeedy ImplemenTation of SNOBOL). A version of SNOBOL that requires less memory and provides faster execution than SNOBOL.

3.282 SPS (Symbolic Programming Systems). A programming language in which terms may represent quantities and locations.

3.283 SPSS (Statistical Package for Social Sciences). A nonprocedural language used for statistical analysis of research results, particularly data collected in polls and surveys.

3.284 SQL (Structured Query Language). A query language designed for accessing data and performing queries on relational databases, standardized by ASC X3.

3.285 STAIRS. A nonprocedural computer language used in manipulation of textual data.

3.286 standard language. Any language that conforms to an existing language standard. For example, ALGOL 60 and ALGOL 68 are considered standard languages.

3.287 STRAND. A concurrent logic programming language.

3.288 stratified language. A language that cannot be used as its own metalanguage. Examples include FORTRAN, COBOL. *Contrast with:* **unstratified language.**
[610.12]

3.289 STRESS (STRUctural Engineering Systems Solver). A problem-oriented programming language used in structural engineering. *See also:* **ICES; STRUDL.**

3.290 structured programming language. A programming language that provides structured program constructs such as single-entry-single-exit sequences, branches, and loops, and facilitates the development of structured programs. *See also:* **block-structured language.**

3.291 STRUDL (STRUctural Design language). An extension of STRESS, used for analysis and design of structures.

3.292 subset. A dialect of a particular language that varies from its referenced standard language such that its capabilities include some, but not all, the capabilities of the referenced language. For example, TINT is a subset of JOVIAL. *Contrast with:* **extension.**

3.293 symbol manipulation language. *See:* **list processing language.**

3.294 symbolic language. (1) A programming language that expresses operations and addresses in symbols convenient to humans rather than in machine language. Examples are assembly language, high-order language. *Contrast with:* **machine language.** [610.12]
(2) *See:* **list processing language.**

3.295 Tell-a-graf. A computer language used to develop presentation and business graphics.

3.296 test language. A computer language used in testing components of hardware or of software. Examples include ATLAS, ATOLL, DETOL, and DMAD.

3.297 TEX (or TeX). A page description language used widely for formatting text containing mathematical symbols.

3.298 text-formatting language. A computer language used to format text documents. Examples include Bookmaster, Cyphertext, DCF, SCRIPT, PAGE, and SCRIBE. *See also:* **page description language.**

3.299 third generation language (3GL). *See:* **high-order language.**

3.300 TINT. A subset of JOVIAL designed for simplified time-sharing programming.

3.301 troff. A text-formatting language used widely in the UNIX environment.

3.302 TROLL. A nonprocedural computer language.

3.303 turtle graphics. *See:* **LOGO.**

3.304 unstratified language. A language that can be used as its own metalanguage; for example English, German. *Contrast with:* **stratified language.** *See also:* **natural language.** [610.12]

3.305 variant. *See:* **dialect.**

3.306 variation. *See:* **dialect.**

3.307 version. *See:* **dialect.**

3.308 VDL (Vienna Definition Language). A metalanguage used to formally define the syntax and semantics of PL/1.

3.309 VDM (Vienna Development Method). A specification language developed by IBM; widely used in Europe.

3.310 VHDL (VHSIC Hardware Description Language). A standard language of the United States Department of Defense, used in the description, design and simulation of very high-speed integrated circuits (VHSIC) and computer logic systems; standardized by IEEE.

3.311 WATFIV (Waterloo FORTRAN V). An extension of WATFOR characterized by additional simplifications in the control (test and looping) constructs and the input/output structure.

3.312 WATFOR (Waterloo FORTRAN). A programming language based on FORTRAN, characterized by its fast compilation, excellent diagnostic messages and debugging aids. *See also:* **WATFIV.**

3.313 YACC (Yet Another Compiler Compiler). A compiler specification language used to express the characteristics of a compiler by defining a set of grammar rules that YACC uses to generate a table-driven lookahead LR(1) (LALR) parser that can be used in a compiler. *See also:* **LEX.**

3.314 Z. A specification language based on typed set theory, developed at Oxford University.

3.315 ZBASIC. A dialect of the BASIC programming language.