

## GLOSSARY

[Source: L.W. Friedman, *Comparative Programming Languages*, Prentice Hall, 1991.]

**Abstract data type.** A programmer-defined data type that specifies a new domain as well as the operations that are allowed on members of that domain. Once defined, the abstract data type may be used like any other type, without regard for the details of how the type has been implemented. See *Data abstraction; Information hiding*.

**Abstraction.** Abstraction is the means by which we consider only such information as is relevant to the problem at hand, ignoring trivial details and unimportant facts. Abstraction models a chosen view of reality in which irrelevant objects or properties are ignored, making the model simpler and more appropriate as an object of study. See *Information hiding*.

**Access-oriented programming paradigm.** In this approach to programming, the access (storing or retrieving) of data objects from variable locations in storage can cause attached procedures to be invoked.

**Access mode.** The mode, or direction, of access from any particular file is determined by whether the file is to be used for input, for output, or both.

**Activation record.** A record, created at the invocation of a subprogram, that contains locally declared variables, subprogram parameters, a pointer to the executable sub-program code, and the return address for the next instruction in sequence in the calling program.

**Actual parameter.** *Argument.* When a subprogram is called, the name of the subprogram is followed by a list that corresponds in number and order to the parameters listed in the subprogram specification. The items in this list, the actual parameters, are specified by the author of the calling program and are the means by which information may be passed to or received from the subprogram.

**Ada.** A programming language for embedded computer systems that was designed according to specifications of the U.S. Department of Defense.

**ALGOL.** ALGORithmic Language. A programming language that was developed to facilitate the communication of ideas among members of the international computing community. Its features include explicit type declarations, arrays with dynamic bounds, recursion, and nesting of program units. Versions include ALGOL58 (originally, IAL), ALGOL60, and ALGOL68.

**ALGOL68.** A much extended revision of ALGOL60, ALGOL68 is a huge, general-purpose language. ALGOL68 innovated the user-defined data type and the pointer type.

**Aliasing.** Aliasing occurs when two variables access the same storage location. Anonymous variable. An unnamed storage location, such as that which holds the result of an evaluated expression.

**APL.** A Programming Language. An applicative, interactive, science-oriented language that assumes the array as the default data structure and features a nonstandard character set.

**Applicative programming paradigm.** See Functional programming paradigm.

**Argument.** See Actual parameter.

**Arithmetic operators.** Operators that act on numeric, computational data.

**Array.** An ordered collection of data elements all of the same type, usually fixed in size. Assembler. A translator program that translates assembly language to machine language. There is a one-to-one correspondence between each assembly-language instruction and the corresponding machine-language instruction.

**Automatic code generator.** A general term to describe any system that accepts high level specifications and outputs a machine-coded program.

**Backward chaining.** One way that rules may be examined by the inference engine of an expert system or rule-based system. In this reasoning mechanism, the system starts with a solution and then tries to prove it correct. See Forward chaining.

**BASIC.** Beginner's All-purpose Symbolic Instructional Code. A high-level language designed to introduce students in nonscientific disciplines to computing. The major goal of the language was to simplify the user interface.

**Batch processing.** A type of processing environment in which there is no interaction of any sort with the executing program. User jobs are submitted in sequential batches.

**Binary search tree.** A binary tree that has been ordered for a binary search, a type of search algorithm that reduces the size of the data set by one-half at each iteration.

**Binary tree.** A type of tree in which each node has at most two descendants.

**Binding.** The association of a property with a particular unit of program code. For example, the binding of one or more attributes (such as location, type, or value) to a variable.

**Binding time.** The time when binding occurs for a particular association. It can occur before run time (early or static binding) or after run time (late or dynamic binding).

**Black Box Model.** A model governing abstraction. In this model, a set of inputs is mapped to a set of outputs. The model does not reveal how the transformation is implemented; only what it does.

**Boolean type.** A binary-valued data type, it may take on such values as (false, true) or (0,1).

**Bottom-up design.** A programming approach in which the programmer begins coding immediately, starting with some low-level subprograms that are well defined. This results in the construction of a large program by building layers on layers of abstraction. This approach, which often results in excessive concern with low-level details, may draw attention away from the larger problem. (See Top-down design for a different approach.)

**Brackets.** A matched pair of symbols or keywords that delimits a portion of code.

**Breadth-first search.** A search strategy employed to search the knowledge base of an expert system. It searches across rules on a level, eliminating those it can, before going on to the next, more detailed level. See Depth-first search.

**C.** A programming language developed at Bell Laboratories, used in coding the routines of the UNIX operating system. It was the first portable operating system. C facilitates the coding of low-level operations in a high level language.

**Call by address.** See Call by location.

**Call by location.** Call by address. Call by reference. The most widely used method of parameter passing. The calling subprogram passes the machine address of the argument to the subprogram. The corresponding parameter is a pointer variable that points to the location of the argument.

**Call by name.** A parameter-passing method in which the evaluation of each argument is deferred until it is actually needed during execution of the subprogram. Instead of passing a value to a parameter, a rule for evaluating the parameter is passed. Each time the parameter name appears in the text of the subprogram, it is "replaced" by the exact text of the argument.

**Call by need.** A parameter-passing method employed by functional programming languages. An argument is evaluated only when its value is needed and not immediately at the point of call.

**Call by reference.** See Call by location.

**Call by result.** A parameter-passing method that is used in the language Ada. Parameters are set up like local variables but are not initialized to any values in the main calling program. At termination, the values stored in the parameters are copied back to the corresponding arguments in the main program.

**Call by value.** A parameter-passing method in which parameters are set up as variables local to the subprogram and are initialized at invocation to the value of the arguments.

**Call by value-result.** A parameter-passing method that was used in ALGOL-W, it combines features of call by value and call by result. A parameter is declared and initialized as a local variable with initial value equal to that of the argument. When the subprogram terminates, the final value of the parameter is passed back to the argument.

**Character type.** A data type that takes on as value a single character, which may be alphabetic, numeric or a special character.

**Class.** A structure that defines the variables and methods for a set of objects that are instances of the class. See Object-oriented programming paradigm.

**COBOL. COmmon Business-Oriented Language.** Designed to meet the needs of the data processing community, COBOL is a language characterized by a heavy reliance on English, a record data structure, file description and manipulation facilities, and noise words for readability.

**Cohesion.** The degree to which each subprogram performs only a single function and all statements contained within that subprogram relate only to that function.

**Compiler.** A type of language processor that transforms an entire source program composed of high-level language statements into an object program consisting of machine-language executable code.

**Compound statement.** A program structure composed of a sequence of two or more simple statements, often delimited by a pair of matching keywords.

**Computer abstraction.** See Virtual computer.

**Concurrent processing.** See parallel processing.

**Constant.** A data item that remains unchanged throughout the execution of the program.

**Constraint-oriented programming paradigm.** In this approach to programming, the programmer specifies a set of relations among a set of data objects. The constraint satisfaction system then attempts to find a solution that satisfies the relations. A spreadsheet is an example of a constraint-oriented program. PROLOG is a special type of constraint-oriented language in that it satisfies logical constraints.

**Control abstraction.** A stylized, structured control construct that enables a programmer to think in terms of high-level concepts as opposed to low-level detailed instructions. For example, constructs of the parallel processing model of subprogram control.

**Coroutines.** Two or more subprograms that exert control over each other.

**Coupling.** The degree of relatedness of a subprogram to the rest of the program.

**Critical region.** In parallel processing, the section of code that contains the statements requiring access to a variable that is shared with other processes.

**Data abstraction.** A language facility that integrates the representation of a programmer- defined data object and the operations that may be performed on that data object into a single syntactic unit. Data abstraction is implemented by information hiding. The object in object-oriented programming is a data abstraction. The data models of database management systems are data abstractions, as are the models for knowledge representation in artificial intelligence applications.

**Database.** A logically interrelated set of data items, related to a particular application or environment, stored on a large-scale direct-access storage device.

**Database management system.** DBMS. Specialized software used to access a database. It is the software interface between the physical storage of the data and the use of the data in various applications.

**Data-centered language.** A data-centered language is one in which programs are developed by means of a formal specification of the data that is to be manipulated. See Process-centered language.

**Data element.** The most basic data entity, the building block with which complicated data structures and large databases are composed.

**Dataflow programming paradigm.** In this approach to programming, the flow of data through a network of operations is specified. Dataflow analysis determines dependencies between data, and the order of execution is in turn determined by these dependencies. The inherent parallelism in the algorithm is exploited. Dataflow programs are mainly intended to run on parallel-architecture dataflow machines, which are still, for the most part, experimental. The dataflow paradigm is one model of parallel processing.

**Data hierarchy.** The logical organization of data stored on permanent storage media such as magnetic tape. Related data items are organized into records and related records are organized into files.

**Data structure.** An organized collection of data objects subject to certain allowable operations.

**Data type.** A domain of data elements associated with a set of operations that act on those elements. Every programming language has some built-in data types.

**DBMS.** See Database management system.

**Deadlock.** When two parallel processes cannot terminate or continue because they remain indefinitely in a state in which each is waiting for the other.

**Declarative programming.** *Nonprocedural programming.* Declarative programming languages have a data-centered approach to programming and are concerned with what is to be done with the data rather than detailing how it is to be accomplished. This term is sometimes applied to a subset of nonprocedural programming languages in which programming is accomplished only by stating facts and assertions.

**Delimiter.** A symbol or keyword that separates pieces of program text. Delimiters provide the punctuation for the program.

**Depth-first search.** A widely used search strategy employed to search the knowledge base of an expert system. It is a search for greater and greater detail along a single search path to a possible goal before attempting to explore another path leading to another goal. See Breadth-first search.

**Deque.** Double-Ended **QUE**ue. A type of queue structure allowing for insertions and deletions at both ends.

**Dimensionality.** The number of dimensions, or directions, of an array structure.

**Direct file.** See Relative file.

**Disabled condition.** An exception that is not being monitored for occurrence.

**Dynamic scoping rule.** In dynamic languages, the scope of a variable is determined during execution time by the most recent occurrence and still active definition of that variable name.

**Efficiency.** A criterion used in evaluating a programming language and its implementation. Refers to the utilization of memory space and processing time.

**Enabled condition.** An exception that is being monitored for occurrence. When the condition occurs, it will raise an exception.

**Entry point.** The point at which execution begins when a subprogram is invoked.

**Enumerated type.** A user-defined type that provides for the enumeration (listing) of the domain of the type by the programmer.

**Exception.** An event that occurs unexpectedly, infrequently, and at random intervals, such as an attempt to divide by zero or a subscript that is out of range.

**Exception handler.** A subprogram, written by the programmer, that interfaces with the operating system and is invoked only when a specified "exceptional condition" (e.g., an attempt to divide by zero) is encountered.

**Exit point.** The point at which execution of the subprogram ends and control is returned to the calling program.

**Expert system shell.** A complete expert system without a knowledge base.

**Explanation facility.** One component of an expert system that can, on demand, justify its conclusions and explain the reasons for its queries. It communicates with the user in the language of the application domain, usually through the user interface.

**Explicit control structure.** A control structure that is imposed by the programmer over the default flow of control provided by the language.

**Expression.** A formula for computing a value, represented as a formalized sequence of operators, operands, and parentheses.

**Extensibility.** A criterion used in evaluating a programming language. An extensible language enables the programmer to define new language components which then become indistinguishable from the language's own built-in primitives.

**External subprogram.** A subprogram that is separate from the text of the program and may be compiled separately into an object module.

**File.** An organized collection of records related to a particular application.

**File organization.** The organization, in terms of a physical ordering, of the records in a file residing on secondary storage, implying the appropriate operations for accessing particular records.

**File structure.** The organization of data residing in secondary storage and subject to input/output related operations.

**Fixed-format language.** A language containing strict rules governing the arrangement of program text on a line.  
**Flowchart.** A pictorial method of expressing program logic flow. Flowcharts are machine and programming language independent and allow programmers to focus on logic flow without having to worry about syntax.

**FLOW-MATIC.** A language designed in 1958 that used English words heavily and was geared to business data

processing. It influenced the subsequent design of COBOL.

**FOCUS.** A full-function 4GT centered on a hierarchical database.

**Formal parameter. Parameter.** Specified in a subprogram, it is a data name place-holder that will be associated with a storage location on invocation of the subprogram. See Actual parameter.

**FORTH.** A language that combines features of both high-level and low-level programming. FORTH arithmetic is based on Reverse Polish notation and programs are constructed using functions.

**FORTRAN.** FORmula TRANslating System. The first high-level programming language, it is imperative and science oriented.

**Forward chaining.** A reasoning mechanism used by the inference engine of an expert system. This reasoning mechanism starts with a premise and uses that as a starting point to examine one item of data at a time and search for a solution. See Back-ward chaining.

**4GT.** 4th Generation Tool. See Full-function 4GT .

**Fourth-generation software.** Declarative programming . Nonprocedural programming. Fourth-generation software is results oriented, concentrating more on what is to be done rather than on describing in detail how to do it. It attempts to maximize human productivity rather than minimize the use of computer time, and it is often aimed at the nontechnical user in a particular application area. Some major trends in fourth-generation software include rule-oriented and object-oriented languages, packaged software, integrated packages, query languages, and integrated programming environments.

**Free-format language.** A free-format language has no fixed fields on the lines of program text. A delimiter is often used to indicate the end of a statement.

**Full-function 4GT.** A full-function 4GT (4th Generation Tool), as well as being centered around a database, will generally have a query language, a report generator, a graphics generator, a facility for screen definition, the ability to do financial and statistical analysis, and it will often have its own procedural or nonprocedural language. FOCUS, NOMAD2, and RAMIS II are examples of full-function 4GTs.

**Function.** A value-returning subprogram-one that returns a single value to the calling program.

**Functional programming paradigm.** Applicative programming paradigm. In a functional programming language, a program is a function that is built up from smaller functions. Functional programming languages achieve results by the application of functions to arguments that may themselves be functions. They are expression oriented rather than statement oriented.

**Generic function.** A function whose type depends on the type of argument used at the point of call.

**Global variable.** A variable that is declared in the main program unit, remains in existence throughout the execution of the program and may be accessed in all program units.

**Goto-less programming.** In theory, all programs can be coded without the unconditional branch, using the structured control constructs of simple sequence, selection, and iteration. Some languages have eliminated the goto entirely.

**Graph.** A collection of nodes and edges. A graph is the most generalized data structure.

**Hierarchy chart.** A graphical view of the entire modular decomposition of a hierarchical program in which every subprogram is represented by a rectangle.

**High-level language.** A programming language that is problem oriented rather than machine oriented. It requires

less detail and fewer lines of code than a low-level language because the language itself handles much of the detail work.

**Identifier.** A programmer-defined word that provides access to program and data entities.

**Imperative.** Statement oriented.

**Imperative programming paradigm.** A statement-oriented approach reflected in all the languages that have their roots in FORTRAN, COBOL, and/or ALGOL. A program is assumed to be a sequence of detailed statements to the computer defining how a task is to be accomplished. Imperative programming languages are characterized by variables, the assignment operation, and the iteration control construct.

**Implicit control.** A subprogram control construct in which a subprogram causes another subprogram to be invoked but does not call it directly, for example, an exception handler.

**Indexed loop.** A form of iteration construct that is used when it is desired to execute a code segment a countable number of times. An index variable is used to control iteration.

**Indexed sequential file.** A file stored sequentially on a direct-access storage device along with an index, which allows for direct access to a particular record in the file.

**Inference engine.** The expert system component governing reasoning.

**Information hiding.** A software engineering concept that a subprogram should have access only to the information that it requires. Any data it does not need to know should be inaccessible to it. Much irrelevant information is thus hidden from the user, protecting the integrity of individual modules and reducing the confusion that comes along with too much information.

**Input parameter.** Values that are input to, and operated on by, the subprogram.

**Input/output parameter.** Values that provide two-way communication between the call-ing subprogram and the called subprogram.

**Integrated package.** A multifunction package of programs containing several commonly used computing applications such as, for personal computers: word processing, spreadsheet, database management system, graphics, and communications. See Package.

**Interactive processing.** A processing environment characterized by interaction between the program and its human user during program execution.

**Internal subprogram.** A subprogram that is part of and contained within the text of the program, and compiled along with the program in which it is called.

**Interpreter.** A type of language processor that operates one instruction at a time, translating a high-level language statement and executing it immediately.

**Iteration.** Repetition. Loop. A control construct that enables repetition of the execution of a group of statements.

**Iterative composition.** See Iteration.

**Job.** A unit that interfaces with the operating system, it may contain several programs and subroutines to be linked together and/or executed in sequence.

**Key.** A data item that serves to uniquely identify a particular record.

**Keyword.** A word that is part of the built-in vocabulary of the programming language.

**Knowledge base.** One component of an expert system, it contains a representation of the expert's knowledge and experience.

**Knowledge engineering.** The process of capturing an expert's knowledge and transforming it into a computer-usable form.

**Label.** An identifier for a program statement.

**Lexical scoping rule.** See Static scoping rule.

**Linked allocation.** One of two ways in which storage can be allocated for most data structures. With linked allocation, each component is a node having two parts: one containing the original data element and the other containing a pointer to the next component. A structure that is allocated storage in a linked manner is a dynamic structure capable of change. Dynamic structures are ordered explicitly, each component containing within itself the location of the next item. See Sequential allocation.

**Linked list.** A dynamic list structure.

**LISP.** An interactive, functional programming language that is used widely in artificial intelligence. Some important features include the use of symbolic expressions as opposed to numbers, the "eval" function for interactive evaluation of LISP statements, and dynamic data structures.

**List.** An ordered sequence of components, which may themselves be lists.

**Local variable.** A variable declared within the confines of the subprogram specification, it is created when the subprogram is invoked and ceases to exist when execution of the subprogram terminates.

**Logic programming paradigm.** A rule-oriented programming approach. The programmer provides a description of a problem in the form of predicate logic statements. The system then is interpreted by a mechanism based on resolution logic. A logic program is composed of: a knowledge base, the program structure, a goal to be proved, and an inference mechanism. This approach does not attempt to deal with uncertainty and does not include an explainer facility as part of its definition. The prototypical logic programming language is PROLOG.

**Logical operator.** An operator that acts on Boolean data and is used in constructing conditions.

**Logo.** A language designed for teaching children mathematics. It is similar to LISP in that it relies on the use of functions and list structures. A distinctive feature of Logo is its "turtle graphics."

**Loop.** See Iteration.

**Low-level language.** A programming language that is machine oriented; i.e., it closely follows the built-in instruction set of the underlying computer. Assembly languages are highly machine oriented.

**Machine code.** The built-in instruction set of the computer, it is formatted in the binary or hexadecimal number system.

**Message passing.** A parallel processing model that simulates a distributed processing system and allows for the exchange of information among communicating processes.

**MODULA-2. MODUlar LAnguage 2.** A high-level language with facilities for system programming, it was intended for use in large-system software design and supports a high degree of problem decomposition and program abstraction.

**Modular programming.** An approach to software engineering in which a large program is designed as a collection of blocks of code. Each subprogram performs a single, clearly defined task and is characterized by a single entry point and a single exit point.

**Modularity.** The degree to which a program is composed of independent subprograms with clearly defined interfaces. See Modular programming.

**Module.** A block of program code. See Subprogram.

**Monitor.** A named subprogram containing a collection of parallel processes that share data.

**Multilinked list.** A list in which each component node contains two or more pointers and is able to reside on several logical lists simultaneously.

**Mutual exclusion.** A problem that must be considered when designing programs with parallel control. When two parallel processes must both be able to access a common resource (e.g., database), but this access must be limited in some way so that the data is protected from an unintentional loss of integrity.

**Named constant.** A word that identifies a constant and is protected from change.

**N-S chart.** Nassi-Shneiderman chart. A method of expressing program logic flow that can be viewed as a pictorial equivalent of pseudo-code. A rectangular, nested flow-chart that "forces" structured control constructs onto the algorithm. Like pseudo-code and flowchart, it is machine- and programming-language independent, and allows programmers to focus on logic flow without having to worry about syntax.

**NATURAL.** A full-function 4GT that is associated with the ADABAS database system.

**NOMAD2.** A full-function 4GT.

**Nonprocedural language.** A language that focuses on the results that one seeks rather than laying out the necessary steps to get there. See Declarative programming, Fourth-generation software.

**Numeric types.** Data types that can accept arithmetic operations. These include integers, floating-point, fixed-point, and complex numbers.

**Object-oriented programming paradigm.** An approach to programming in which the organizing principle is the data object. The data object is an active entity asked to perform operations (called methods) on itself. A complex problem is viewed as a network of objects that communicate with each other by message passing that invokes these methods.

**Object program.** A machine-executable program produced as output from a compiler.

**Operand.** A component of an expression, an operand represents access to a data value and may be a constant, a variable name, a function reference, or another expression.

**Operation.** A single computer instruction; a type of simple subprogram.

**OPS5.** Official Production System, version 5. A rule-oriented production-system language used in artificial intelligence.

**Output parameter.** An output parameter receives its value as the result of processing done within the subprogram. Its value is passed back to the point of call.

**Package.** A program or set of programs designed to solve a problem common to a large number of users-for example, statistical analysis or graphics.

**Parallel processing.** Concurrent processing. The concurrent execution of two or more subprograms called processes. They operate simultaneously, relatively independently, and may communicate with each other and share common resources.

**Parameter.** See Formal parameter.

**Pascal.** A narrowly defined, simple and elegant programming language designed by Niklaus Wirth in his opposition to ALGOL68. Pascal influenced the design of many subsequent languages.

**PL/1. Programming Language/1.** A general, multipurpose language developed by IBM that drew on concepts from several major languages including FORTRAN and COBOL. Some important features include multitasking, defaults, and explicit use of pointers and list processing.

**Pointer.** A reference to an unnamed data object.

**Portability.** A criterion used in evaluating a programming language. If a language is portable, then programs written in the language will work and provide similar results on many different machine/compiler; i.e., they will be machine independent.

**Precedence rules.** Rules that govern the implicit control over the operations within an expression.

**Problem oriented.** The degree to which a program can handle a particular problem or application area efficiently. For example, COBOL is particularly well suited for problems in business data processing; FORTRAN is especially appropriate for problems in the scientific areas. PL/1 is an example of a true general-purpose language, since it can handle any problem area as well as another.

**Procedural language.** A process-centered language. The primary objective of a procedural programming language is to express the individual detailed instructions that the computer will follow in order to accomplish its task. Languages differ in the degree to which they are procedural. See Nonprocedural language.

**Process-centered language.** A language that is concerned with the details of computing a solution and with the process of problem decomposition. See Data-centered language .

**Processing environment.** The degree to which the program interacts with its environment during execution time. Some examples of processing environments are: batch processing, interactive processing, and parallel processing.

**Processing mode.** The manner in which a file will be used- for example, batch mode or query mode.

**Production-system programming paradigm.** In this rule-oriented approach to programming, a program consists of a set of condition-action rules called productions. Production systems can handle uncertainty and are frequently used in the construction of expert systems.

**Productivity tool.** A program or set of programs that speeds up the rate of application development-for example, report writer, program generator, natural language front-end, graphics package.

**Program abstraction.** A relatively independent program unit for the processing of a specific task that may be called by another subprogram without the programmer needing to know exactly how the program abstraction accomplishes its goal, only what the function is that it performs. See Abstraction, Information hiding.

**Programming environment.** A set of tools that includes a programming language and serves to aid in program development. Some of the tools that may be contained within a programming environment, also known as a software development environment, include an editor, a linker, debugger, and an on-line help facility.

**Programming language.** A notation for expressing data structures and instructions for computing purposes.

**Programming myopia.** A condition characterized by dependence on one particular programming language.

**Programming paradigm.** An approach to programming. Most programming languages assume a particular programming paradigm as the major organizing principle. The most prevalent paradigm today is the imperative

programming paradigm.

**PROLOG.** **PRO**gramming in LOGic. A logic-oriented language that is declarative rather than procedural. It is used in artificial intelligence including the areas of natural language processing and the building of expert systems.

**Pseudocode.** A nonpictorial method of expressing program logic flow. Pseudo-code is machine and programming language independent and allows the programmer to focus on logic flow without having to worry about syntax.

**QBE.** Query By Example. A popular query language that is easy to use because it works with a visual format.

**Queue.** A list into which new items may be inserted at one end only (the REAR), and from which items may be deleted at the other end only (the FRONT).

**Query language.** A user-friendly language that usually accompanies a database management system and is normally powerful enough to allow for sophisticated programming of a procedural or nonprocedural nature. Examples are NATURAL for ADABAS, SQL, and QBE. PROLOG is sometimes considered a query language based on logical relations.

**Raising an exception.** This involves noticing that the exception condition has occurred, causing program execution to be interrupted or temporarily suspended, and the invocation of the exception handler.

**RAMIS II.** Rapid Access Management Information System II. The first commercially available, full-function 4GT.

**Random file.** See Relative file.

**Readability.** A criterion used in evaluating a programming language. The degree to which programs written in that language may be easily read and understood by humans.

**Real-time processing.** A processing environment in which external processes, such as physical objects, interact with and impose strict time constraints on the responses from executing programs.

**Reasoning mechanism.** The control structure of the inference engine of an expert system, which attempts to mimic the way people reason. Two reasoning mechanisms are backward chaining and forward chaining.

**Record.** A hierarchical data structure. An ordered collection of data elements not necessarily of the same type, each element identified by name.

**Recursion.** A programming technique in which a subprogram calls itself. See Recursion.

**Recursive subprogram.** A subprogram that calls itself.

**Reengineering.** A concept that program design and modification are both made at the specification level. Thus, changes to the specification effect changes to the code.

**Reference.** An evaluated variable name that provides access to data.

**Relational operators.** Operators that form conditions and can operate on virtually all types of data.

**Relative file.** Direct file. Random file. An unordered collection of component records stored on a direct access storage device in which each record may be accessed directly by location (absolute or relative).

**Repetition.** See Iteration.

**Repository.** An extended data dictionary used by CASE tools. It contains definitions for every object in the system and for relationships among objects. It also keys these objects to the actual program code. Objects may be, for example, definitions for screen or printed reports, database schemata, structure charts, dataflow diagrams.

**Reserved words.** Language keywords that are off-limits to the programmer and may not be used to denote programmer-supplied words.

**Rule-based programming paradigm.** See Production-system programming paradigm.

**Rule-oriented programming paradigms.** An approach to programming in which one may: specify facts and rules about objects and their relationships; query the system regarding these objects and relationships; combine facts to express them as a single rule; easily integrate new facts and rules into a program. PROLOG and OPS5 are rule-oriented languages that represent two divergent world views: the logic programming and production-system programming paradigms.

**Scheduled call.** A type of subprogram control used in event-oriented simulation programs. Control over some subprograms is managed by a timing mechanism (a subprogram that may be built into the programming language or coded by the programmer), and a subprogram is scheduled to be invoked at a particular activation time.

**Scope of a variable.** The portion of the program in which a variable name is associated with and accesses a particular value.

**Search strategy.** The technique used to search the knowledge base of an expert system. See Depth-first search, Breadth-first search.

**Selection.** A control construct that is employed when one wishes to choose among two or more alternative blocks of code.

**Selective composition.** See Selection.

**Self-documenting.** A criterion used in evaluating a programming language. The degree to which a program is self-explaining. Self-documenting techniques enhance the readability of a program.

**Semaphores.** A Boolean variable with two operations: wait and signal. It is used to control access to a shared resource in parallel processing.

**Separator.** A delimiter that indicates the end of one program entity and the beginning of the next.

**Sequential allocation.** One of two ways in which storage can be allocated for most data structures. With sequential allocation, a single block of contiguous storage locations is allocated for the structure. A structure-allocated storage in a sequential manner is incapable of change throughout its lifetime; it may be called a static structure. See Linked allocation.

**Sequential file.** A linear sequence of related records that may be ordered on some key field.

**Sequential composition.** See Simple sequence.

**Set.** A data structure of unspecified size containing an unordered collection of distinct values.

**Side effect.** A change to a global variable, occurring as the result of processing done in a subprogram.

**Simple call.** A highly constrained subprogram control construct that allows the calling subprogram to exert total control over the called subprogram.

**Simple sequence.** The simplest control structure and the default in imperative languages. Individual statements are executed one after another in the order in which they appear in the program.

**Simple statement.** A statement that contains a single action, for example, ADD.

**SIMULA. SIMULATION Language.** A programming language for simulation applications. SIMULA67, the 1967 revision, was of a more general nature.

**Smalltalk.** The language that defines the object-oriented approach to programming. It is a language embedded within an environment. The environment is a user-friendly, interactive interface with multiple overlapping windows, graphical and textual menus, and a mouse for selecting and pointing.

**SNOBOL. StriNg Oriented SymBolic Language.** A string-processing language for formula manipulation.

**Software engineering.** The discipline of software engineering can be traced back to the structured programming techniques of the early 1970s. It includes techniques for analysis, design, testing, validation, and maintenance of software.

**Source program.** A program written in a high-level language and then input to the compile process.

**SQL. Structured Query Language.** A query language that is becoming the standard way to communicate with relational databases.

**Stack.** An ordered set of items into which new items may be inserted, and from which items may be deleted at one end only (the TOP).

**Static scoping rule.** Lexical scoping rule. Every variable is accessible in the subprogram in which it is declared and to any subprogram nested inside of it-unless the same variable name is re-declared as a local variable in some inner subprogram-but is not accessible to any subprogram outside or surrounding it.

**Static structure.** A structure that is incapable of change throughout its lifetime.

**Stepwise refinement.** The consecutive specification of subprograms over several, progressively more detailed levels of abstraction. This process of stepwise refinement results in the modular decomposition of a program into a hierarchical structure.

**Storage structure.** What data structures become after being mapped to memory.

**String.** An ordered sequence of characters of dynamically changing size.

**Strongly typed language.** A programming language in which the types of all variables are determined at compile time. Usually includes rigorous type checking.

**Structured control constructs.** Statement-level control structures are either (low-level) stylized sequences of instructions or high-level language statements, each with a single entry point and a single exit point, which fall into one of the following three categories: simple sequence, selection, iteration. (The goto is not a structured control construct.)

**Structured English.** See Pseudocode .

**Structured programming.** A set of programming techniques for implementing program and control abstractions in a hierarchical manner. Its purpose is to make programs more readable, less error-prone, and easier to maintain. Three techniques in this category are: modularity, top-down design, and the use of structured control constructs.

**Subprogram.** A block of code that has a name, is relatively independent, and performs a specific task.

**Subprogram call.** The invocation of a subprogram.

**Subprogram library.** A collection of external subprograms that may contain such built-in functions and/or procedures as arithmetic functions, statistical procedures, searching/sorting arrays, etc. This allows programmers to build on rather than duplicate the work of previous programmers.

**Subtype.** This user-defined type allows for the specification of the domain as a subrange of another already existing type.

**Symbol table.** The symbol table maps program symbols such as variables with their properties such as type, size, and relative location.

**Synchronization.** A problem that must be considered when designing programs with parallel control. When concurrently executing processes attempt to communicate with each other, one process must wait for the completion of or a signal from another process.

**Test-after loop.** A form of iteration in which statements contained within the body of the loop will always be executed at least once regardless of whether the condition tested is true or false the first time around.

**Test-before loop.** A form of iteration in which statements in the body of the loop are iterated as long as a particular condition, which is tested before each iteration, is true.

**Textfile.** A stream of characters, separated into lines by an end-of-line character such as the carriage-return.

**Top-down design.** A structured programming technique in which a large, complex task is structured into a hierarchy of manageable subtasks. At each level in the hierarchy, low-level decisions and detail work are delayed to be performed later when work on a lower level will be considered.

**Tree.** A hierarchical collection of nodes and the branches connecting them. Each node may be considered the "root" of another tree.

**Tree traversal.** The process of "traveling" over the branches of a tree in such a way so that each node is accessed only once.

**Typing system.** A facility for defining new data types and for declaring variables to be of such types.

**User-defined type.** A data type that is defined by a programmer and provides a limited means for specifying the domain of a "new" type based on some other predefined type.

**User interface.** An interactive, often graphics-based system component that facilitates communication between the computer system and the human user.

**Variable.** A receptacle for storing data values.

**Variable-length record.** A record with a changeable number of repeated fields.

**Variant record.** A record that is allowed to have two or more variations.

**Virtual computer.** The abstract machine that the user sees and interacts with, for example a "COBOL computer." The same hardware with different sets of software can be transformed into different virtual computers.

**von Neumann computer.** The architecture on which most of today's computers are based. It uses a configuration in which the central processor is tied very closely to an internal memory unit containing a large number of unique addressable storage locations, which may be retrieved one at a time, and a one-word bus for transporting data between these two components. One instruction is executed at a time, in sequence.

**Writability.** A criterion used in evaluating a programming language, it refers to the degree of easiness to design, code, test, run, document, and modify programs.