IEEE
Std 610.12-1990
(Revision and redesignation of IEEE Std 792-1983)

IEEE Standard Glossary of Software Engineering Terminology

Sponsor

Standards Coordinating Committee of the Computer Society of the IEEE

Approved September 28, 1990

IEEE Standards Board

Abstract: IEEE Std 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology*, identifies terms currently in use in the field of Software Engineering. Standard definitions for those terms are established.
Keywords: Software engineering; glossary; terminology; definitions; dictionary

ISBN 1-55937-067-X

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

fault tolerance. (1) The ability of a system or component to continue normal operation despite the presence of hardware or software faults. *See also:* **error tolerance; fail safe; fail soft; fault secure; robustness.**
(2) The number of faults a system or component can withstand before normal operation is impaired.
(3) Pertaining to the study of errors, faults, and failures, and of methods for enabling systems to continue normal operation in the presence of faults. *See also:* **recovery; redundancy; restart.**

fault tolerant. Pertaining to a system or component that is able to continue normal operation despite the presence of faults.

FCA. Acronym for **functional configuration audit.**

feasibility. The degree to which the requirements, design, or plans for a system or component can be implemented under existing constraints.

feature. (IEEE Std 1008-1987 [10]) *See:* **software feature.**

fetch. To locate and load computer instructions or data from storage. *See also:* **move; store.**

fifth generation language (5GL). A computer language that incorporates the concepts of knowledge-based systems, expert systems, inference engines, and natural language processing. *Contrast with:* **assembly language; fourth generation language; high order language; machine language.** *Note:* Specific languages are defined in P610.13 [17].

figurative constant. A data name that is reserved for a specific constant in a programming language. For example, the data name THREE may be reserved to represent the value 3. *See also:* **literal.**

file. A set of related records treated as a unit. For example, in stock control, a file could consist of a set of invoice records.

finite state machine. A computational model consisting of a finite number of states and transitions between those states, possibly with accompanying actions.

firmware. The combination of a hardware device and computer instructions and data that reside as read-only software on that device. *Notes:* (1) This term is sometimes used to refer only to the hardware device or only to the computer instructions or data, but these meanings are deprecated. (2) The confusion surrounding this term has led some to suggest that it be avoided altogether.

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

flag. A variable that is set to a prescribed state, often "true" or "false," based on the results of a process or the occurrence of a specified condition. *See also:* **indicator; semaphore.**

flexibility. The ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed. *Syn:* **adaptability.** *See also:* **extendability; maintainability.**

flow diagram. *See:* **flowchart.**

flow of control. *See:* **control flow.**

flowchart (flow chart). A control flow diagram in which suitably annotated geometrical figures are used to represent operations, data, or equipment, and arrows are used to indicate the sequential flow from one to another. *Syn:* **flow diagram.** *See also:* **block diagram; box diagram; bubble chart; graph; input-process-output chart; structure chart.**

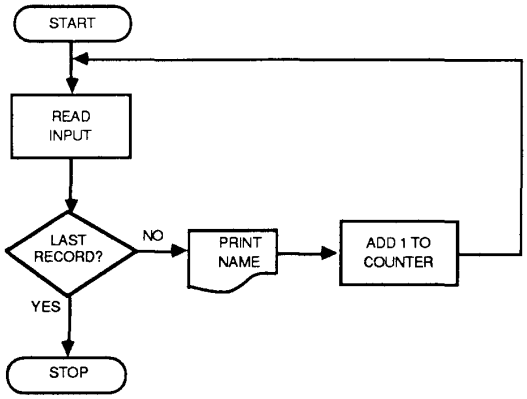

Fig 10
Flowchart

call for data and the instant at which the transfer of data is started.

lateral compression. In software design, a form of demodularization in which two or more modules that execute one after the other are combined into a single module. *Contrast with:* **downward compression; upward compression.**

leading decision. A loop control that is executed before the loop body. *Contrast with:* **trailing decision.** *See also:* **WHILE.**

library. *See:* **software library.**

licensing standard. (IEEE Std 1002-1987 [9]) A standard that describes the characteristics of an authorization given by an official or a legal authority to an individual or organization to do or own a specific thing.

life cycle. *See:* **software life cycle; system life cycle.**

link. (1) To create a load module from two or more independently translated object modules or load modules by resolving cross-references among them. *See also:* **linkage editor.**
(2) A part of a computer program, often a single instruction or address, that passes control and parameters between separate modules of the program. *Syn:* **linkage.**
(3) To provide a link as in (2).

linkage. *See:* **link (2).**

linkage editor. A computer program that creates a single load module from two or more independently translated object modules or load modules by resolving cross-references among the modules and, possibly, by relocating elements. May be part of a loader. *Syn:* **linker.** *See also:* **linking loader.**

linker. *See:* **linkage editor.**

linking loader. A computer program that reads one or more object modules into main memory in preparation for execution, creates a single load module by resolving cross-references among the separate modules, and, in some cases, adjusts the addresses to reflect the storage locations into which the code has been loaded. *See also:* **absolute loader; relocating loader; linkage editor.**

list. (1) A set of data items, each of which has the same data definition.
(2) To print or otherwise display a set of data items.
Note: IEEE Std 610.5-1990 [2] defines Data Management terms.

list processing language. A programming language designed to facilitate the manipulation of data expressed in the form of lists. Examples are LISP and IPL. *See also:* **algebraic language; algorithmic language; logic programming language.**

listing. An ordered display or printout of data items, program statements, or other information.

literal. In a source program, an explicit representation of the value of an item; for example, the word FAIL in the instruction: If $x = 0$ then print "FAIL". *See also:* **immediate data; figurative constant.**

load. (1) To read machine code into main memory in preparation for execution and, in some cases, to perform address adjustment and linking of modules. *See also:* **loader.**
(2) To copy computer instructions or data from external storage to internal storage or from internal storage to registers. *Contrast with:* **store (2).** *See also:* **fetch; move.**

load-and-go. An operating technique in which there are no stops between the loading and execution phases of a computer program.

load map. A computer-generated list that identifies the location or size of all or selected parts of memory-resident code or data.

load module. A computer program or subprogram in a form suitable for loading into main storage for execution by a computer; usually the output of a linkage editor. *See also:* **object module.**

device; usually measured in words or bytes. *See also:* **channel capacity; memory capacity.**

storage efficiency. The degree to which a system or component performs its designated functions with minimum consumption of available storage. *See also:* **execution efficiency.**

store. (1) To place or retain data in a storage device.
(2) To copy computer instructions or data from a register to internal storage or from internal storage to external storage. *Contrast with:* **load (2).** *See also:* **fetch; move.**

straight-line code. A sequence of computer instructions in which there are no loops.

straight-line coding. A programming technique in which loops are avoided by stating explicitly and in full all of the instructions that would be involved in the execution of each loop. *See also:* **unwind.**

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

stress testing. Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements. *See also:* **boundary value.**

strong typing. A feature of some programming languages that requires the type of each data item to be declared, precludes the application of operators to inappropriate data types, and prevents the interaction of data items of incompatible types.

structural testing. Testing that takes into account the internal mechanism of a system or component. Types include branch testing, path testing, statement testing. *Syn:* **glass-box testing; white-box testing.** *Contrast with:* **functional testing (1).**

structure chart. A diagram that identifies modules, activities, or other entities in a system or computer program and shows how larger or more general entities break down into smaller, more specific entities. *Note:* The result is not necessarily the same as that shown in a call graph. *Syn:* **hierarchy chart; program structure chart.** *Contrast with:* **call graph.**

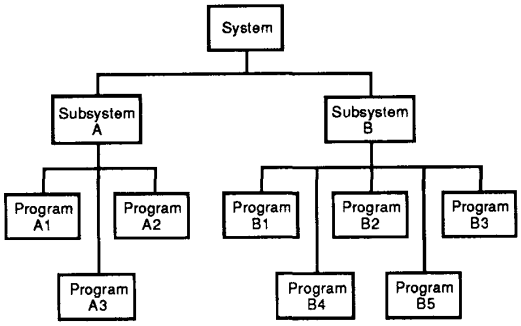

Fig 16
Structure Chart

structure clash. In software design, a situation in which a module must deal with two or more data sets that have incompatible data structures. *See also:* **data structure-centered design; order clash.**

structured design. (1) Any disciplined approach to software design that adheres to specified rules based on principles such as modularity, top-down design, and stepwise refinement of data, system structures, and processing steps. *See also:* **data structure-centered design; input-process-output; modular decomposition; object-oriented design; rapid prototyping; stepwise refinement; transaction analysis; transform analysis.**
(2) The result of applying the approach in (1).

structured program. A computer program constructed of a basic set of control structures, each having one entry and one exit. The set of control structures typically includes: sequence of two or more instructions, conditional selection of one of two or more sequences of instructions, and repetition of a sequence of instructions. *See also:* **structured design.**

structured programming. Any software development technique that includes struc-