

# **IEEE Standard Computer Dictionary**

## **A Compilation of IEEE Standard Computer Glossaries**

# **610**

Sponsor  
**Standards Coordinating Committee  
of the  
IEEE Computer Society**

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

**Keywords:** Glossary; terminology; definitions; dictionary.

Library of Congress Catalog Number 90-086306

ISBN 1-55937-079-3

Copyright ©1990 by

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

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

- tolerance discipline. In common usage, the terms "error" and "bug" are used to express this meaning. *See also: data-sensitive fault; program sensitive fault; equivalent faults; fault masking; intermittent fault.* [610.12]
- fault dictionary.** A list of faults in a system or component, and the tests that have been designed to detect them. [610.12]
- fault masking.** A condition in which one fault prevents the detection of another. [610.12]
- fault secure.** Pertaining to a system or component in which no failures are produced from a prescribed set of faults. *See also: fault tolerance; fail safe; fail soft.* [610.12]
- fault seeding.** *See: error seeding.* [610.12]
- fault tolerance.** (1) The ability of a system or component to continue normal operation despite the presence of hardware or software faults. *See also: error tolerance; fail safe; fail soft; fault secure; robustness.* [610.12]  
 (2) The number of faults a system or component can withstand before normal operation is impaired. [610.12]  
 (3) Pertaining to the study of errors, faults, and failures, and of methods for enabling systems to continue normal operation in the presence of faults. *See also: recovery (2); redundancy; restart.* [610.12a]
- fault tolerant.** Pertaining to a system or component that is able to continue normal operation despite the presence of faults. [610.12]
- fax.** Abbreviation for **facsimile.** [610.2a]
- FCA.** Acronym for **functional configuration audit.** [610.12]
- FCFS.** Acronym for **first-come, first-served.** *See: first-in, first-out.* [610.5]
- FE.** Acronym for **format effector character.** [610.5]
- feasibility.** The degree to which the requirements, design, or plans for a system or component can be implemented under existing constraints. [610.12]
- feature.** (1) In pattern recognition, an attribute of a pattern that may contribute to pattern classification; for example, size, texture, or shape. [610.4]  
 (2) *See: software feature.* [610.128]
- feature extraction.** A step in pattern recognition, in which measurements or observations are processed to find attributes that can be used to assign patterns to pattern classes. [610.4]
- feature space.** In pattern recognition, a set of all possible  $n$ -tuples  $(x_1, x_2, \dots, x_n)$  that can be used to represent  $n$  features of a pattern. *See also: measurement space.* [610.4]
- feedback.** That portion of the output of a control system used as input for another phase of the system, particularly for self-correcting, self-regulating, or control purposes, as in closed-loop control. [610.2]
- feedback control.** *See: closed-loop control.* [610.2]
- FEFO.** Acronym for **first-ended, first-out.** [610.5]
- fetch.** To locate and load computer instructions or data from storage. *See also: move; store.* [610.12]
- FF.** Acronym for **form feed character.** [610.5]
- Fibonacci number.** An integer in the Fibonacci series. [610.1]
- Fibonacci search.** A dichotomizing search in which, at each step in the search, the set of items is partitioned in accordance with the Fibonacci series. For example, a set of 8 items is partitioned to 5 and 3, the subset of 5 is partitioned to 3 and 2, and so on. If the number of items in the original set is other than a Fibonacci number, the next higher Fibonacci number is used to partition the set. *Contrast with: binary search; interpolation search.* [610.5]
- Fibonacci series.** A series of integers formulated by the Italian mathematician Leonardo Fibonacci, in which each integer is equal to the sum of the two preceding

- list sorting.** A sorting technique in which the items to be sorted form a linked list and the links between the items in the list are manipulated in such a way that, in the final list, the items form a linked list in sorted order. *See also:* **address table sorting; key sorting.** [610.5]
- list structure.** (1) A list, each item of which is either a single data item or a list structure itself. [610.5]  
(2) A data structure that contains one or more lists. *Syn:* **compound list.** [610.5]
- literal.** (1) Composed of characters, as in a literal variable name used to contain a customer's name. [610.5]  
(2) In a source program, an explicit representation of the value of an item; for example, the word FAIL in the instruction: If  $x = 0$  then print "FAIL". *See also:* **immediate data; figurative constant.**[610.12]
- load.** (1) To read machine code into main memory in preparation for execution and, in some cases, to perform address adjustment and linking of modules. *See also:* **loader.** [610.12]  
(2) To copy computer instructions or data from external storage to internal storage or from internal storage to registers. *Contrast with:* **store** (2). *See also:* **fetch; move.** [610.12]  
(3) To insert data values into a database that previously contained no data. *Syn:* **populate.** *See also:* **download; upload.** [610.5]
- load-and-go.** An operating technique in which there are no stops between the loading and execution phases of a computer program. [610.12]
- loaded origin.** The address of the initial storage location of a computer program at the time the program is loaded into main memory. *Contrast with:* **assembled origin.** *See also:* **offset** (1); **starting address.** [610.12]
- loader.** (1) A computer program that reads machine code into main memory in preparation for execution and, in some cases, adjusts the addresses and links the modules. Types include absolute loader, linking loader, relocating loader. *See also:* **bootstrap; linkage editor.** [610.12]  
(2) Any program that reads programs or data into main memory. [610.12]
- loading factor.** (1) The maximum amount of usable space in a physical block after accounting for block overhead. [610.5]  
(2) The ratio of the number of stored entities in a file to the maximum number of entries that can be stored in a unit of data medium. [610.5]
- load map.** A computer-generated list that identifies the location or size of all or selected parts of memory-resident code or data. [610.12]
- load module.** A computer program or subprogram in a form suitable for loading into main storage for execution by a computer; usually the output of a linkage editor. *See also:* **object module.** [610.12]
- local compaction.** In microprogramming, compaction in which microoperations are not moved beyond the boundaries of the single entry, single exit sequential blocks in which they occur. *Contrast with:* **global compaction.** [610.12]
- local data.** Data that can be accessed by only one module or set of nested modules in a computer program. *Contrast with:* **global data.** [610.12]
- local variable.** A variable that can be accessed by only one module or set of nested modules in a computer program. *Contrast with:* **global variable.** [610.12]
- lock.** (1) To exclude users from updating data that is being updated by another user. *Note:* Depending on the implementation, locking may occur on a field, record or an entire file. *See also:* **deadlock; exclusive lock.** [610.5]  
(2) To exclude users from accessing data. *Syn:* **field-locking; file-locking; record-locking.** [610.5]
- locking.** In code extension characters, having the characteristic that a change in interpretation applies to all coded representations following, or to all coded representations of a given class, until the next appropriate code

- storage allocation.** An element of computer resource allocation, consisting of assigning storage areas to specific jobs and performing related procedures, such as transfer of data between main and auxiliary storage, to support the assignments made. *See also:* **buffer; contiguous allocation; cyclic search; memory compaction; overlay; paging; virtual storage.** [610.12]
- storage breakpoint.** *See:* **data breakpoint.** [610.12]
- storage capacity.** The maximum number of items that can be held in a given storage device; usually measured in words or bytes. *See also:* **channel capacity; memory capacity.** [610.12]
- storage efficiency.** The degree to which a system or component performs its designated functions with minimum consumption of available storage. *See also:* **execution efficiency.** [610.12]
- storage location.** An area in a storage device that can be explicitly and uniquely specified by means of an address. [610.5]
- storage schema.** In a CODASYL database, statements expressed in data storage definition language that describe storage areas, stored records, and any associated indices and access paths supporting the records and sets defined by a given schema. *See also:* **CODASYL database.** [610.5]
- storage stack.** *See:* **stack.** [610.5]
- storage structure.** (1) The manner in which data structures are represented in storage. [610.5]  
 (2) The configuration of a database resident on computer storage devices after mapping the data elements of the logical structure of the database onto their respective physical counterparts. *Note:* The relationships and associations that provide the physical means for accessing the information stored in the database are preserved. [610.5]
- store.** (1) To place or retain data in a storage device. [610.5a, 610.12]  
 (2) To copy computer instructions or data from a register to internal storage or from internal storage to external storage. *Contrast with:* **load (2); retrieve. *See also:* **fetch; move.** [610.5a, 610.12]**
- stored paragraph.** *See:* **boilerplate text.** [610.2]
- stored record.** *See:* **internal record.** [610.5]
- straight binary.** *See:* **binary.** [610.1]
- straight insertion sort.** *See:* **insertion sort.** [610.5]
- straight-line code.** A sequence of computer instructions in which there are no loops. [610.12]
- straight-line coding.** A programming technique in which loops are avoided by stating explicitly and in full all of the instructions that would be involved in the execution of each loop. *See also:* **unwind.** [610.12]
- straight line sort.** *See:* **linear sort.** [610.5]
- straight radix sort.** A radix sort in which items are sorted repeatedly on successive digits within the numeric representation of the sort key, starting with the least significant digit. [610.5]
- straight selection sort.** *See:* **selection sort.** [610.5]
- straight two-way merge sort.** A variation of the natural two-way merge sort in which the set to be sorted is repeatedly divided into two ordered subsets of length 2 to the power of  $k$ , where  $k$  is the number of passes made so far. *Contrast with:* **natural two-way merge sort.** [610.5]
- stratified language.** A language that cannot be used as its own metalanguage. Examples include FORTRAN, COBOL. *Contrast with:* **unstratified language.** [610.12]
- stress testing.** Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements. *See also:* **boundary value.** [610.12]
- string.** (1) A sequence of bits, characters, or other entities; for example, the bit string