

software engineering glossary

Editor: Richard H. Thayer ■ California State Univ., Sacramento ■ thayer@csus.edu
Technical Reviewer: Merlin Dorfman ■ Cisco Systems ■ dorfman@computer.org

Software Design, Part 3

object-oriented design: A software development technique in which a system or component is expressed in terms of objects and connections between those objects. [IEEE Std 610.12-1990] *Contrast with function-oriented design and data-structure-oriented design.*

OOD: Acronym for object-oriented design.

point design: The selection of one design that satisfies the requirements without examining other, potentially more effective designs.

preliminary design: *See architectural design.*

program design language (PDL): A specification language with special constructs and sometimes verification protocols, used to develop, analyze, and document a program design. *See also pseudocode.* [IEEE Std 610.12-1990]

pseudocode: A general term for structured English or program design language.

requirements traceability tool: A software development tool that establishes a traceability among itemized software requirements specifications, design elements, code elements, and test cases. It also supports various associated query, analysis, and report-generation capabilities.

reverse engineering: A software engineering approach that derives a system's design or requirements from its code. The design might be represented by a program design language (PDL) or a formal language. By creating a logical view from a physical system, reverse engineering helps maintain and enhance existing systems.

SDD: Acronym for software design document.

software design: The use of scientific principles, technical information, and imagination in the definition of a software system to perform prespecified functions with maximum economy and efficiency. *See also design.*

software design audit: A review of a software product to determine compliance with requirements, standards, and contractual documents.

software design concept: A fundamental idea (such as information hiding) that can be applied to designing a system.

software design description: 1. A representation of software created to facilitate analysis, planning, implementation, and decision making. The software design description serves as a medium for communicating software design information and can be thought of as a blueprint or model of the system. [IEEE Std 1012-1986] 2. A representation of a software system created to facilitate analysis, planning, implementation, and decision making. A blueprint or model of the software system. The SDD serves as the primary medium for communicating software design information. *See also detailed design description.* [IEEE Std 1016-1987]

software design document: *See software design description.* [IEEE Std 610.12-1990]

software design notation: A means of describing a software design. It can be diagrammatic, symbolic, or textual. For example, structure charts and pseudocode are software design notations. *Also*

called software design representation.

software design requirement: *See design constraint.*

software design review: A formal meeting at which a system's preliminary or detailed design is presented to the user, customer, or other interested parties for comment and approval. [ANSI/IEEE Std 610.12-1990]

software design specification: *See software design description.*

software design strategy: The overall plan and direction for performing design. For example, functional decomposition is a software design strategy. *See also design strategy.*

software design verification: The evaluation of a design to determine correctness with respect to stated requirements, conformance to design standards, system efficiency, and other criteria.

structured design: A disciplined approach to software design that adheres to a specified set of rules based on principles such as top-down design, stepwise refinement, and dataflow analysis.

test design specification: A document specifying the details of the test approach for a software feature or combination of software features and identifying the associated tests. [ANSI/IEEE Std 829-1983]

top-down design: The process of designing a system by identifying its major components, decomposing them into their low-level components, and iterating until the desired level of detail is achieved. *Contrast with bottom-up design.* ☞