

Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Their jobs usually involve: Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language. It is usually easier to code in "high-level" languages than in "low-level" ones. Allen Downey, in his book *How To Think Like A Computer Scientist*, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Programs were mostly entered using punched cards or paper tape. Use of a static code analysis tool can help detect some possible problems. Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference.