

This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Scripting and breakpointing is also part of this process. Scripting and breakpointing is also part of this process. Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. It is usually easier to code in "high-level" languages than in "low-level" ones. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. Code-breaking algorithms have also existed for centuries. 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. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. Normally the first step in debugging is to attempt to reproduce the problem. Techniques like Code refactoring can enhance readability. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Use of a static code analysis tool can help detect some possible problems. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research. It affects the aspects of quality above, including portability, usability and most importantly maintainability. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. 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.