

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. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Integrated development environments (IDEs) aim to integrate all such help. 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. 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). They are the building blocks for all software, from the simplest applications to the most sophisticated 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. However, readability is more than just programming style. 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. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. 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. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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 of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. 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. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. 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. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.