

Programs were mostly entered using punched cards or paper tape. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Techniques like Code refactoring can enhance readability. 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. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Programmable devices have existed for centuries. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Ideally, the programming language best suited for the task at hand will be selected. Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. 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. 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. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. 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. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. Computer programmers are those who write computer software. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.