

Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Programs were mostly entered using punched cards or paper tape. Following a consistent programming style often helps readability. Use of a static code analysis tool can help detect some possible problems. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages. Scripting and breakpointing is also part of this process. 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. There exist a lot of different approaches for each of those tasks. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. 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. New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation). Ideally, the programming language best suited for the task at hand will be selected. Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. There exist a lot of different approaches for each of those tasks. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks.