

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). Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. However, readability is more than just programming style. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. It is usually easier to code in "high-level" languages than in "low-level" ones. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. It affects the aspects of quality above, including portability, usability and most importantly maintainability. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software. 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. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. 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. Integrated development environments (IDEs) aim to integrate all such help. It is very difficult to determine what are the most popular modern programming languages. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Programs were mostly entered using punched cards or paper tape. These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.