

Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. 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. 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Computer programmers are those who write computer software. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. 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. 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). It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash. One approach popular for requirements analysis is Use Case analysis. Different programming languages support different styles of programming (called programming paradigms). Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. Programmable devices have existed for centuries. 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. The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine. 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 text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. 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.