

After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Use of a static code analysis tool can help detect some possible problems. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. Scripting and breakpointing is also part of this process. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. 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. 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. It is usually easier to code in "high-level" languages than in "low-level" ones. Unreadable code often leads to bugs, inefficiencies, and duplicated code. 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. 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. Following a consistent programming style often helps readability. It is usually easier to code in "high-level" languages than in "low-level" ones. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Code-breaking algorithms have also existed for centuries. 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. There exist a lot of different approaches for each of those tasks.