

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. Programmable devices have existed for centuries. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Code-breaking algorithms have also existed for centuries. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. 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. 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. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. 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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. However, readability is more than just programming style. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. One approach popular for requirements analysis is Use Case analysis. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. 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.