

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. Also, specific user environment and usage history can make it difficult to reproduce the problem. It is very difficult to determine what are the most popular modern programming languages. Programmable devices have existed for centuries. Also, specific user environment and usage history can make it difficult to reproduce the problem. Programmable devices have existed for centuries. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. It is very difficult to determine what are the most popular modern programming languages. Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language. It is usually easier to code in "high-level" languages than in "low-level" ones. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. 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. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. 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. However, readability is more than just programming style. Programmable devices have existed for centuries. 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. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. There are many approaches to the Software development process. 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. 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.