

They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Allen Downey, in his book *How To Think Like A Computer Scientist*, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Also, specific user environment and usage history can make it difficult to reproduce the problem. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Unreadable code often leads to bugs, inefficiencies, and duplicated code. 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. Ideally, the programming language best suited for the task at hand will be selected. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. It is very difficult to determine what are the most popular modern programming languages. Programmable devices have existed for centuries. Many applications use a mix of several languages in their construction and use. It is usually easier to code in "high-level" languages than in "low-level" ones. Different programming languages support different styles of programming (called programming paradigms). It is usually easier to code in "high-level" languages than in "low-level" ones. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Scripting and breakpointing is also part of this process. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. 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.