

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. Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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. 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. One approach popular for requirements analysis is Use Case analysis. 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. 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. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. 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. 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. 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. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. There exist a lot of different approaches for each of those tasks. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams.