

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. Different programming languages support different styles of programming (called programming paradigms). Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. 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. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. There are many approaches to the Software development process. As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the *Book of Ingenious Devices*. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. It is very difficult to determine what are the most popular modern programming languages. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Scripting and breakpointing is also part of this process. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. It is usually easier to code in "high-level" languages than in "low-level" ones. Programming languages are essential for software development. 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. Use of a static code analysis tool can help detect some possible problems. 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. However, readability is more than just programming style. 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. 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. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se.