

Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. Following a consistent programming style often helps readability. 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. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. 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. New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation). 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. Integrated development environments (IDEs) aim to integrate all such help. Their jobs usually involve: Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language. 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. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Many applications use a mix of several languages in their construction and use. 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. However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages. Scripting and breakpointing is also part of this process. Many applications use a mix of several languages in their construction and use. Programming languages are essential for software development. Integrated development environments (IDEs) aim to integrate all such help. Also, specific user environment and usage history can make it difficult to reproduce the problem. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. 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. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design 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.