

There exist a lot of different approaches for each of those tasks. 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 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. 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 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem. Unreadable code often leads to bugs, inefficiencies, and duplicated code. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Unreadable code often leads to bugs, inefficiencies, and duplicated code. 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Integrated development environments (IDEs) aim to integrate all such help. Programmable devices have existed for centuries. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Normally the first step in debugging is to attempt to reproduce the problem. Normally the first step in debugging is to attempt to reproduce the problem. 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. 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. 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).