Different programming languages support different styles of programming (called programming paradigms). Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Different programming languages support different styles of programming (called programming paradigms). Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. It is very difficult to determine what are the most popular modern programming languages. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Also, specific user environment and usage history can make it difficult to reproduce the problem. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Techniques like Code refactoring can enhance readability. Computer programmers are those who write computer software. 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. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. 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. 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. 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. Scripting and breakpointing is also part of this process. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. There exist a lot of different approaches for each of those tasks. 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. However, readability is more than just programming style. For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash. Whatever the approach to development may be, the final program must satisfy some fundamental properties. These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.