One approach popular for requirements analysis is Use Case analysis. 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. However, readability is more than just programming style. 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. Code-breaking algorithms have also existed for centuries. Following a consistent programming style often helps readability. Programming languages are essential for software development. 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. 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. 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. Programmable devices have existed for centuries. Computer programmers are those who write computer software. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Scripting and breakpointing is also part of this process. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). 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. It is very difficult to determine what are the most popular modern programming languages. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Programmable devices have existed for centuries. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.