In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. 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. Different programming languages support different styles of programming (called programming paradigms). Different programming languages support different styles of programming (called programming paradigms). However, Charles Babbage had already written his first program for the Analytical Engine in 1837. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. Programming languages are essential for software development. Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. 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. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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. Ideally, the programming language best suited for the task at hand will be selected. 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. There are many approaches to the Software development process. 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. For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software. Ideally, the programming language best suited for the task at hand will be selected.