Techniques like Code refactoring can enhance readability. Different programming languages support different styles of programming (called programming paradigms). It is very difficult to determine what are the most popular modern programming languages. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. It is very difficult to determine what are the most popular modern programming languages. 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. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). One approach popular for requirements analysis is Use Case analysis. Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. Code-breaking algorithms have also existed for centuries. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. Programs were mostly entered using punched cards or paper tape. 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). Ideally, the programming language best suited for the task at hand will be selected. Scripting and breakpointing is also part of this process. Use of a static code analysis tool can help detect some possible problems. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Many applications use a mix of several languages in their construction and use. 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. Use of a static code analysis tool can help detect some possible problems. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Techniques like Code refactoring can enhance readability.