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. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. It is usually easier to code in "high-level" languages than in "low-level" ones. 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). Following a consistent programming style often helps readability. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Use of a static code analysis tool can help detect some possible problems. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. There exist a lot of different approaches for each of those tasks. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. One approach popular for requirements analysis is Use Case analysis. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. Computer programmers are those who write computer software. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. 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. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. Programs were mostly entered using punched cards or paper tape. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Different programming languages support different styles of programming (called programming paradigms). 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.