This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. 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. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. 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. 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. There are many approaches to the Software development process. 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. However, readability is more than just programming style. 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. However, readability is more than just programming style. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Ideally, the programming language best suited for the task at hand will be selected. Use of a static code analysis tool can help detect some possible problems. 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. 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. Normally the first step in debugging is to attempt to reproduce the problem. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. One approach popular for requirements analysis is Use Case analysis.