It is very difficult to determine what are the most popular modern programming languages. 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. One approach popular for requirements analysis is Use Case analysis. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Scripting and breakpointing is also part of this process. Unreadable code often leads to bugs, inefficiencies, and duplicated code. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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. Different programming languages support different styles of programming (called programming paradigms). High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. Programmable devices have existed for centuries. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Use of a static code analysis tool can help detect some possible problems. However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. Normally the first step in debugging is to attempt to reproduce the problem. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Also, specific user environment and usage history can make it difficult to reproduce the problem.