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.  
 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.  
 Programs were mostly entered using punched cards or paper tape.  
While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se.  
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.  
 It is very difficult to determine what are the most popular modern programming languages.  
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.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
 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).  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
 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.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.