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.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
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.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
 It is very difficult to determine what are the most popular modern programming languages.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
 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).  
The purpose of programming is to find a sequence of instructions that will automate the performance of a task (which can be as complex as an operating system) on a computer, often for solving a given problem.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.