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..  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
 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.  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
 Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute.  
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.  
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.  
Integrated development environments (IDEs) aim to integrate all such help.  
There are many approaches to the Software development process.  
Ideally, the programming language best suited for the task at hand will be selected.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
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.