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..  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference.  
Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
 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.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
  
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.  
 Following a consistent programming style often helps readability.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
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.  
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.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.