Programming languages are essential for software development..  
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.  
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.  
 Programmable devices have existed for centuries.  
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.  
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.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
Their jobs usually involve:  
 Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language.  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
Integrated development environments (IDEs) aim to integrate all such help.  
 Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
 Different programming languages support different styles of programming (called programming paradigms).  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.