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.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
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, readability is more than just programming style.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
There are many approaches to the Software development process.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Techniques like Code refactoring can enhance readability.  
There exist a lot of different approaches for each of those tasks.  
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.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
 Programs were mostly entered using punched cards or paper tape.