Programs were mostly entered using punched cards or paper tape..  
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.  
Ideally, the programming language best suited for the task at hand will be selected.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Some of these factors include:  
 The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Techniques like Code refactoring can enhance readability.  
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.  
One approach popular for requirements analysis is Use Case analysis.  
Many applications use a mix of several languages in their construction and use.  
However, readability is more than just programming style.