Ideally, the programming language best suited for the task at hand will be selected..  
 Code-breaking algorithms have also existed for centuries.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
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.  
Scripting and breakpointing is also part of this process.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
One approach popular for requirements analysis is Use Case analysis.  
Use of a static code analysis tool can help detect some possible problems.  
There are many approaches to the Software development process.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
Techniques like Code refactoring can enhance readability.  
Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.  
 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.  
However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.