Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code..  
 It is very difficult to determine what are the most popular modern programming languages.  
 Code-breaking algorithms have also existed for centuries.  
Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
Many applications use a mix of several languages in their construction and use.  
Normally the first step in debugging is to attempt to reproduce the problem.  
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.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
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.  
One approach popular for requirements analysis is Use Case analysis.  
Ideally, the programming language best suited for the task at hand will be selected.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.