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 usually easier to code in "high-level" languages than in "low-level" ones.  
There are many approaches to the Software development process.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
  
However, readability is more than just programming style.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.  
For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
 Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code.  
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.