Techniques like Code refactoring can enhance readability.  
One approach popular for requirements analysis is Use Case analysis.  
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.  
Also, those involved with software development may at times engage in reverse engineering, which is the practice of seeking to understand an existing program so as to re-implement its function in some way.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
The following properties are among the most important:  
  
 In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
This is interpreted into 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.  
Programming involves tasks such as analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms (usually in a particular programming language, commonly referred to as coding).  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
This is interpreted into machine code.  
Ideally, the programming language best suited for the task at hand will be selected.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.