Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability..  
 Following a consistent programming style often helps readability.  
One approach popular for requirements analysis is Use Case analysis.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
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.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
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.  
Their jobs usually involve:  
 Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
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.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
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.  
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.  
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.  
Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.