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 programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years.  
One approach popular for requirements analysis is Use Case analysis.  
Integrated development environments (IDEs) aim to integrate all such help.  
There exist a lot of different approaches for each of those tasks.  
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.  
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.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
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.  
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).  
Many applications use a mix of several languages in their construction and use.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
 Programs were mostly entered using punched cards or paper tape.