The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference.  
The source code of a program is written in one or more languages that are intelligible to programmers, rather than machine code, which is directly executed by the central processing unit.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
There are many approaches to the Software development process.  
This is interpreted into machine code.  
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.  
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.  
There exist a lot of different approaches for each of those tasks.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
 Different programming languages support different styles of programming (called programming paradigms).  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.