It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages..  
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.  
 It is very difficult to determine what are the most popular modern programming languages.  
Programming languages are essential for software development.  
There exist a lot of different approaches for each of those tasks.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
Scripting and breakpointing is also part of this process.  
One approach popular for requirements analysis is Use Case analysis.  
Many applications use a mix of several languages in their construction and use.  
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.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.