Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic 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.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.  
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.  
Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit.  
However, readability is more than just programming style.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
One approach popular for requirements analysis is Use Case analysis.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
 It is very difficult to determine what are the most popular modern programming languages.  
 Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code.  
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.