For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software..  
 Allen Downey, in his book How To Think Like A Computer Scientist, writes:  
 Many computer languages provide a mechanism to call functions provided by shared libraries.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
Normally the first step in debugging is to attempt to reproduce the problem.  
One approach popular for requirements analysis is Use Case analysis.  
Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL).  
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.  
 New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation).  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.