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..  
 Different programming languages support different styles of programming (called programming paradigms).  
 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).  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists.  
 Computer programmers are those who write computer software.  
One approach popular for requirements analysis is Use Case analysis.  
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.  
 Programmable devices have existed for centuries.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
Normally the first step in debugging is to attempt to reproduce the problem.  
 Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display.  
 Code-breaking algorithms have also existed for centuries.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.