Many applications use a mix of several languages in their construction and use..  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
 Code-breaking algorithms have also existed for centuries.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
The following properties are among the most important:  
  
 In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code.  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
 The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
 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.