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.  
To produce machine code, the source code must either be compiled or transpiled.  
One approach popular for requirements analysis is Use Case analysis.  
However, readability is more than just programming style.  
Integrated development environments (IDEs) aim to integrate all such help.  
However, readability is more than just programming style.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
 These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.  
The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference.  
In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams.  
One approach popular for requirements analysis is Use Case analysis.