Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages..  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
One approach popular for requirements analysis is Use Case analysis.  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).  
Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.  
Techniques like Code refactoring can enhance readability.  
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.  
 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.  
 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.  
While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se.  
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.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
 Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code.  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.