One approach popular for requirements analysis is Use Case analysis.  
However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
Techniques like Code refactoring can enhance readability.  
Also, those involved with software development may at times engage in reverse engineering, which is the practice of seeking to understand an existing program so as to re-implement its function in some way.  
 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.  
This is interpreted into machine code.  
Relatedly, software engineering combines engineering techniques and principles with software development.  
Programming languages are essential for software development.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
Programming involves tasks such as analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms (usually in a particular programming language, commonly referred to as coding).  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.