Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years..  
 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.  
Many applications use a mix of several languages in their construction and use.  
Some of these factors include:  
 The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills.  
 Computer programmers are those who write computer software.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
Ideally, the programming language best suited for the task at hand will be selected.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
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.  
Normally the first step in debugging is to attempt to reproduce the problem.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.