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..  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
 Computer programmers are those who write computer software.  
Integrated development environments (IDEs) aim to integrate all such help.  
 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.  
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.  
 Following a consistent programming style often helps readability.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
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.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.