Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards..  
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.  
There exist a lot of different approaches for each of those tasks.  
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.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
 New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation).  
Programming languages are essential for software development.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
Integrated development environments (IDEs) aim to integrate all such help.  
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.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
There are many approaches to the Software development process.  
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.  
 Allen Downey, in his book How To Think Like A Computer Scientist, writes:  
 Many computer languages provide a mechanism to call functions provided by shared libraries.