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..  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute.  
For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Techniques like Code refactoring can enhance readability.  
 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).  
For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.  
Integrated development environments (IDEs) aim to integrate all such help.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
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.