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.  
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.  
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).  
Techniques like Code refactoring can enhance readability.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Ideally, the programming language best suited for the task at hand will be selected.  
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.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
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.