Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic..  
Programming languages are essential for software development.  
Techniques like Code refactoring can enhance readability.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
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.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
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 first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.  
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.  
Their jobs usually involve:  
 Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language.  
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.  
One approach popular for requirements analysis is Use Case analysis.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
Integrated development environments (IDEs) aim to integrate all such help.