Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
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).  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Techniques like Code refactoring can enhance readability.  
Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL).  
This is interpreted into machine code.  
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.  
To produce machine code, the source code must either be compiled or transpiled.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
 Programmable devices have existed for centuries.  
Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
 Different programming languages support different styles of programming (called programming paradigms).  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.