Different programming languages support different styles of programming (called programming paradigms)..  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
 Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code.  
There exist a lot of different approaches for each of those tasks.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
 Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line.  
 Following a consistent programming style often helps readability.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
Programming languages are essential for software development.  
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.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
 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.  
Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.