Scripting and breakpointing is also part of this process..  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
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.  
Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit.  
 These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.  
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).  
Programming languages are essential for software development.  
Integrated development environments (IDEs) aim to integrate all such help.  
 It is very difficult to determine what are the most popular modern programming languages.  
Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
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.  
Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists.