Different programming languages support different styles of programming (called programming paradigms)..  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.  
Programming languages are essential for software development.  
Many applications use a mix of several languages in their construction and use.  
 Computer programmers are those who write computer software.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
Ideally, the programming language best suited for the task at hand will be selected.  
 Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display.  
Integrated development environments (IDEs) aim to integrate all such help.  
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.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 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.