Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages..  
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.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
 Following a consistent programming style often helps readability.  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.  
 Different programming languages support different styles of programming (called programming paradigms).  
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.  
 It is very difficult to determine what are the most popular modern programming languages.  
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.  
Integrated development environments (IDEs) aim to integrate all such help.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.