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.  
The source code of a program is written in one or more languages that are intelligible to programmers, rather than machine code, which is directly executed by the central processing unit.  
However, while these might be considered part of the programming process, often the term software development is more likely used for this larger overall process – whereas the terms programming, implementation, and coding tend to be focused on the actual writing of code.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
Ideally, the programming language best suited for the task at hand will be selected.  
Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.  
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.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
Normally the first step in debugging is to attempt to reproduce the problem.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
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.