Relatedly, software engineering combines engineering techniques and principles with software development.  
Techniques like Code refactoring can enhance readability.  
Scripting and breakpointing is also part of this process.  
Ideally, the programming language best suited for the task at hand will be selected.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
The purpose of programming is to find a sequence of instructions that will automate the performance of a task (which can be as complex as an operating system) on a computer, often for solving a given problem.  
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.  
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.  
Scripting and breakpointing is also part of this process.  
 Code-breaking algorithms have also existed for centuries.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Programming involves tasks such as analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms (usually in a particular programming language, commonly referred to as coding).  
 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.  
Also, those involved with software development may at times engage in reverse engineering, which is the practice of seeking to understand an existing program so as to re-implement its function in some way.  
  
 Computer programming is the process of performing particular computations (or more generally, accomplishing specific computing results), usually by designing and building executable computer programs.