Relatedly, software engineering combines engineering techniques and principles with software development.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
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.  
Normally the first step in debugging is to attempt to reproduce the problem.  
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).  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research.  
This is interpreted into machine code.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
The following properties are among the most important:  
  
 In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code.  
To produce machine code, the source code must either be compiled or transpiled.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
 Code-breaking algorithms have also existed for centuries.