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.  
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.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
There are many approaches to the Software development process.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
To produce machine code, the source code must either be compiled or transpiled.  
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.  
For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash.  
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.  
Ideally, the programming language best suited for the task at hand will be selected.  
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.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.