However, readability is more than just programming style.  
Ideally, the programming language best suited for the task at hand will be selected.  
To produce machine code, the source code must either be compiled or transpiled.  
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.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices.  
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.  
In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams.  
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.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 It is very difficult to determine what are the most popular modern programming languages.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
Many applications use a mix of several languages in their construction and use.  
One approach popular for requirements analysis is Use Case analysis.