Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability..  
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.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
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.  
 Code-breaking algorithms have also existed for centuries.  
 These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.  
 Different programming languages support different styles of programming (called programming paradigms).  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
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.  
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.  
However, readability is more than just programming style.  
There are many approaches to the Software development process.  
Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic.  
 The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems.