Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
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.  
Use of a static code analysis tool can help detect some possible problems.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
However, readability is more than just programming style.  
Scripting and breakpointing is also part of this process.  
For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
  
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 can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code.