Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
Programming languages are essential for software development.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Some of these factors include:  
 The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills.  
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.  
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.  
Also, those involved with software development may at times engage in reverse engineering, which is the practice of seeking to understand an existing program so as to re-implement its function in some way.  
There exist a lot of different approaches for each of those tasks.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Ideally, the programming language best suited for the task at hand will be selected.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
 Tasks accompanying and related to programming include testing, debugging, source code maintenance, implementation of build systems, and management of derived artifacts, such as the machine code of computer programs.  
 Tasks accompanying and related to programming include testing, debugging, source code maintenance, implementation of build systems, and management of derived artifacts, such as the machine code of computer programs.