By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
Techniques like Code refactoring can enhance readability.  
Scripting and breakpointing is also part of this process.  
However, readability is more than just programming style.  
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.  
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.  
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.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
This is interpreted into machine code.  
 Following a consistent programming style often helps readability.  
 Programmable devices have existed for centuries.  
Programming involves tasks such as analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms (usually in a particular programming language, commonly referred to as coding).  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 Computer programmers are those who write computer software.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.