Compiling takes the source code from a low-level programming language and converts it into machine code.  
Ideally, the programming language best suited for the task at hand will be selected.  
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 are many approaches to the Software development process.  
This is interpreted into machine code.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
There exist a lot of different approaches for each of those tasks.  
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.  
There exist a lot of different approaches for each of those tasks.  
Integrated development environments (IDEs) aim to integrate all such help.  
 Following a consistent programming style often helps readability.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.