Techniques like Code refactoring can enhance readability.  
Normally the first step in debugging is to attempt to reproduce the problem.  
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.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software.  
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.  
Programming languages are essential for software development.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
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).  
Many applications use a mix of several languages in their construction and use.  
There exist a lot of different approaches for each of those tasks.  
Integrated development environments (IDEs) aim to integrate all such help.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.