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.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
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.  
Integrated development environments (IDEs) aim to integrate all such help.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
There are many approaches to the Software development process.  
Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute.  
Techniques like Code refactoring can enhance readability.  
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.  
One approach popular for requirements analysis is Use Case analysis.  
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.  
Use of a static code analysis tool can help detect some possible problems.  
 It is very difficult to determine what are the most popular modern programming languages.