Unreadable code often leads to bugs, inefficiencies, and duplicated code..  
One approach popular for requirements analysis is Use Case analysis.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
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.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
Ideally, the programming language best suited for the task at hand will be selected.  
Use of a static code analysis tool can help detect some possible problems.  
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.  
However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory.  
Many applications use a mix of several languages in their construction and use.  
 Programs were mostly entered using punched cards or paper tape.  
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.  
For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash.  
 Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line.