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..  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.  
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.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
However, readability is more than just programming style.  
Scripting and breakpointing is also part of this process.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
One approach popular for requirements analysis is Use Case analysis.  
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.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
Many applications use a mix of several languages in their construction and use.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
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.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.