New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation)..  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
One approach popular for requirements analysis is Use Case analysis.  
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.  
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.  
  
 Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks.  
 It is very difficult to determine what are the most popular modern programming languages.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
Scripting and breakpointing is also part of this process.  
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.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
 Following a consistent programming style often helps readability.