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)..  
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.  
 Different programming languages support different styles of programming (called programming paradigms).  
Programming languages are essential for software development.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
 These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
 Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code.  
 Allen Downey, in his book How To Think Like A Computer Scientist, writes:  
 Many computer languages provide a mechanism to call functions provided by shared libraries.  
Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
 It is very difficult to determine what are the most popular modern programming languages.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
The following properties are among the most important:  
  
 In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code.