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..  
 Programs were mostly entered using punched cards or paper tape.  
 Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code.  
Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL).  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.  
 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.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic.  
There exist a lot of different approaches for each of those tasks.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit.  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
 Different programming languages support different styles of programming (called programming paradigms).