Also, specific user environment and usage history can make it difficult to reproduce the problem..  
Use of a static code analysis tool can help detect some possible problems.  
Integrated development environments (IDEs) aim to integrate all such help.  
Techniques like Code refactoring can enhance readability.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
 Code-breaking algorithms have also existed for centuries.  
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.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
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.  
  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 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.  
Normally the first step in debugging is to attempt to reproduce the problem.  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.