Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Many applications use a mix of several languages in their construction and use.  
Integrated development environments (IDEs) aim to integrate all such help.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Also, those involved with software development may at times engage in reverse engineering, which is the practice of seeking to understand an existing program so as to re-implement its function in some way.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
Many applications use a mix of several languages in their construction and use.  
Many applications use a mix of several languages in their construction and use.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
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.  
 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.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
This is interpreted into machine code.  
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.