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.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
 Different programming languages support different styles of programming (called programming paradigms).  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
 Programs were mostly entered using punched cards or paper tape.  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
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.  
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.  
 Following a consistent programming style often helps readability.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
 Code-breaking algorithms have also existed for centuries.  
Techniques like Code refactoring can enhance readability.  
There exist a lot of different approaches for each of those tasks.  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).