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.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
 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.  
 Programmable devices have existed for centuries.  
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.  
There are many approaches to the Software development process.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists.  
 Code-breaking algorithms have also existed for centuries.  
Normally the first step in debugging is to attempt to reproduce the problem.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input.