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.  
Normally the first step in debugging is to attempt to reproduce the problem.  
The source code of a program is written in one or more languages that are intelligible to programmers, rather than machine code, which is directly executed by the central processing unit.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
The purpose of programming is to find a sequence of instructions that will automate the performance of a task (which can be as complex as an operating system) on a computer, often for solving a given problem.  
However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
There exist a lot of different approaches for each of those tasks.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
Scripting and breakpointing is also part of this process.  
Programming languages are essential for software development.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
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.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine.