Use of a static code analysis tool can help detect some possible problems.  
One approach popular for requirements analysis is Use Case analysis.  
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.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Normally the first step in debugging is to attempt to reproduce the problem.  
To produce machine code, the source code must either be compiled or transpiled.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
Normally the first step in debugging is to attempt to reproduce the problem.  
FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research.  
Scripting and breakpointing is also part of this process.  
Relatedly, software engineering combines engineering techniques and principles with software development.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
However, readability is more than just programming style.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
 Code-breaking algorithms have also existed for centuries.