Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
Techniques like Code refactoring can enhance readability.  
Many applications use a mix of several languages in their construction and use.  
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.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
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.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Scripting and breakpointing is also part of this process.  
There are many approaches to the Software development process.  
However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
Use of a static code analysis tool can help detect some possible problems.  
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.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.