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.  
Many applications use a mix of several languages in their construction and use.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
However, while these might be considered part of the programming process, often the term software development is more likely used for this larger overall process – whereas the terms programming, implementation, and coding tend to be focused on the actual writing of code.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
One approach popular for requirements analysis is Use Case analysis.  
Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
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.  
Also, those involved with software development may at times engage in reverse engineering, which is the practice of seeking to understand an existing program so as to re-implement its function in some way.  
The following properties are among the most important:  
  
 In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code.  
There exist a lot of different approaches for each of those tasks.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
 It is very difficult to determine what are the most popular modern programming languages.