Use of a static code analysis tool can help detect some possible problems.  
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.  
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.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
However, readability is more than just programming style.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
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.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
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.  
Some of these factors include:  
 The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills.  
Many applications use a mix of several languages in their construction and use.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
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.