However, readability is more than just programming style.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Programming languages are essential for software development.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
This is interpreted into machine code.  
One approach popular for requirements analysis is Use Case analysis.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
Ideally, the programming language best suited for the task at hand will be selected.  
For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
 It is very difficult to determine what are the most popular modern programming languages.  
In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams.  
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.  
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.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.