For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input..  
 These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.  
 Programmable devices have existed for centuries.  
 Different programming languages support different styles of programming (called programming paradigms).  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
 Computer programmers are those who write computer software.  
Techniques like Code refactoring can enhance readability.  
Integrated development environments (IDEs) aim to integrate all such help.  
Scripting and breakpointing is also part of this process.  
For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software.  
There are many approaches to the Software development process.  
 The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems.  
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.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
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.