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..  
One approach popular for requirements analysis is Use Case analysis.  
 Code-breaking algorithms have also existed for centuries.  
 Programmable devices have existed for centuries.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
Integrated development environments (IDEs) aim to integrate all such help.  
 Different programming languages support different styles of programming (called programming paradigms).  
 It is very difficult to determine what are the most popular modern programming languages.  
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.  
Techniques like Code refactoring can enhance readability.  
 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.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
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.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.