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..  
 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.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
 Programmable devices have existed for centuries.  
However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory.