High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware..  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
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.  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Techniques like Code refactoring can enhance readability.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
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.  
 New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation).  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.