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..  
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.  
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.  
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.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
 Programmable devices have existed for centuries.  
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.  
Techniques like Code refactoring can enhance readability.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
 Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Programming languages are essential for software development.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.