However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
Programming languages are essential for software development.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
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.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists.  
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.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
One approach popular for requirements analysis is Use Case analysis.