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.  
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.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
Techniques like Code refactoring can enhance readability.  
FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research.  
Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years.  
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.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
One approach popular for requirements analysis is Use Case analysis.  
Programming languages are essential for software development.  
There exist a lot of different approaches for each of those tasks.  
This is interpreted into machine code.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.