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.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.  
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.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
There exist a lot of different approaches for each of those tasks.  
In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
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.  
 Programs were mostly entered using punched cards or paper tape.  
Use of a static code analysis tool can help detect some possible problems.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
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.