Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
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.  
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.  
Scripting and breakpointing is also part of this process.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
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.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
Techniques like Code refactoring can enhance readability.  
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.  
There are many approaches to the Software development process.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
There are many approaches to the Software development process.  
One approach popular for requirements analysis is Use Case analysis.