The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging)..  
Integrated development environments (IDEs) aim to integrate all such help.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Scripting and breakpointing is also part of this process.  
Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
One approach popular for requirements analysis is Use Case analysis.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
Programming languages are essential for software development.  
In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.  
For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software.