Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards..  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
 Computer programmers are those who write computer software.  
While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se.  
Techniques like Code refactoring can enhance readability.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
Use of a static code analysis tool can help detect some possible problems.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
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.  
One approach popular for requirements analysis is Use Case analysis.  
For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash.  
 New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation).  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.