Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards..  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
Scripting and breakpointing is also part of this process.  
Use of a static code analysis tool can help detect some possible problems.  
There are many approaches to the Software development process.  
 Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display.  
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.  
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.  
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.  
When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear.  
 Code-breaking algorithms have also existed for centuries.  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.  
 Computer programmers are those who write computer software.  
One approach popular for requirements analysis is Use Case analysis.