High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware..  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
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.  
There are many approaches to the Software development process.  
Normally the first step in debugging is to attempt to reproduce the problem.  
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.  
 These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.  
 Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
  
 Following a consistent programming style often helps readability.  
 Computer programmers are those who write computer software.