Many applications use a mix of several languages in their construction and use..  
 Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line.  
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.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 Code-breaking algorithms have also existed for centuries.  
Normally the first step in debugging is to attempt to reproduce the problem.  
Techniques like Code refactoring can enhance readability.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
There are many approaches to the Software development process.  
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.  
 In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.  
 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.  
As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices.  
 It is very difficult to determine what are the most popular modern programming languages.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.