It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages..  
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.  
  
Programming languages are essential for software development.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Scripting and breakpointing is also part of this process.  
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.  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.  
Ideally, the programming language best suited for the task at hand will be selected.  
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.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
 Allen Downey, in his book How To Think Like A Computer Scientist, writes:  
 Many computer languages provide a mechanism to call functions provided by shared libraries.  
 Code-breaking algorithms have also existed for centuries.  
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.  
Integrated development environments (IDEs) aim to integrate all such help.