The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'..  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
Techniques like Code refactoring can enhance readability.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
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.  
 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.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
 Code-breaking algorithms have also existed for centuries.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Use of a static code analysis tool can help detect some possible problems.