Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation..  
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.  
 Programmable devices have existed for centuries.  
 Programs were mostly entered using punched cards or paper tape.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
 Following a consistent programming style often helps readability.  
 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.  
 The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
There exist a lot of different approaches for each of those tasks.  
Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists.  
Use of a static code analysis tool can help detect some possible problems.  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
 Computer programmers are those who write computer software.  
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.