It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages..  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.  
One approach popular for requirements analysis is Use Case analysis.  
Normally the first step in debugging is to attempt to reproduce the problem.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.  
However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
 Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code.  
 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.  
For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
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.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Integrated development environments (IDEs) aim to integrate all such help.