It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages..  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
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.  
The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
 New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation).  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
Ideally, the programming language best suited for the task at hand will be selected.  
 Different programming languages support different styles of programming (called programming paradigms).  
There exist a lot of different approaches for each of those tasks.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.