Programmable devices have existed for centuries..  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
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.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
However, readability is more than just programming style.  
 Code-breaking algorithms have also existed for centuries.  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
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.  
Programming languages are essential for software development.  
Ideally, the programming language best suited for the task at hand will be selected.  
Normally the first step in debugging is to attempt to reproduce the problem.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash.