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..  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.  
 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.  
Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
Normally the first step in debugging is to attempt to reproduce the problem.  
Techniques like Code refactoring can enhance readability.  
There exist a lot of different approaches for each of those tasks.  
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.  
Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
One approach popular for requirements analysis is Use Case analysis.  
Many applications use a mix of several languages in their construction and use.  
Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.