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..  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
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.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Use of a static code analysis tool can help detect some possible problems.  
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.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Programming languages are essential for software development.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
Techniques like Code refactoring can enhance readability.  
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.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
 Allen Downey, in his book How To Think Like A Computer Scientist, writes:  
 Many computer languages provide a mechanism to call functions provided by shared libraries.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
  
The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'.