The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'..  
 It is very difficult to determine what are the most popular modern programming languages.  
 Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
 Following a consistent programming style often helps readability.  
 Code-breaking algorithms have also existed for centuries.  
However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory.  
 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.  
However, readability is more than just programming style.  
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.  
 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.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.