Also, specific user environment and usage history can make it difficult to reproduce the problem.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
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.  
However, readability is more than just programming style.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software.  
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.  
The source code of a program is written in one or more languages that are intelligible to programmers, rather than machine code, which is directly executed by the central processing unit.  
 Tasks accompanying and related to programming include testing, debugging, source code maintenance, implementation of build systems, and management of derived artifacts, such as the machine code of computer programs.  
One approach popular for requirements analysis is Use Case analysis.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.