Unreadable code often leads to bugs, inefficiencies, and duplicated code..  
 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.  
One approach popular for requirements analysis is Use Case analysis.  
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.  
 Following a consistent programming style often helps readability.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
However, readability is more than just programming style.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Use of a static code analysis tool can help detect some possible problems.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
 It is very difficult to determine what are the most popular modern programming languages.  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.