Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications..  
However, readability is more than just programming style.  
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.  
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.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
 Following a consistent programming style often helps readability.  
One approach popular for requirements analysis is Use Case analysis.  
Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.  
Integrated development environments (IDEs) aim to integrate all such help.  
Ideally, the programming language best suited for the task at hand will be selected.  
As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices.  
 New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation).  
 Programmable devices have existed for centuries.  
In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams.