There are many approaches to the Software development process.  
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.  
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.  
The purpose of programming is to find a sequence of instructions that will automate the performance of a task (which can be as complex as an operating system) on a computer, often for solving a given problem.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
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.  
Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL).  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
 Different programming languages support different styles of programming (called programming paradigms).  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
  
 Computer programming is the process of performing particular computations (or more generally, accomplishing specific computing results), usually by designing and building executable computer programs.  
One approach popular for requirements analysis is Use Case analysis.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
 It is very difficult to determine what are the most popular modern programming languages.  
Compiling takes the source code from a low-level programming language and converts it into machine code.