They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
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.  
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.  
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.  
Programming languages are essential for software development.  
Transpiling on the other hand, takes the source-code from a high-level programming language and converts it into bytecode.  
Compiling takes the source code from a low-level programming language and converts it into machine code.  
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.  
FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research.  
Scripting and breakpointing is also part of this process.  
The following properties are among the most important:  
  
 In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.  
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.