They are the building blocks for all software, from the simplest applications to the most sophisticated ones..  
There exist a lot of different approaches for each of those tasks.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit.  
 Computer programmers are those who write computer software.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
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.  
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.  
 Programs were mostly entered using punched cards or paper tape.  
 Different programming languages support different styles of programming (called programming paradigms).  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).  
 Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code.