

A computer program is a collection of instructions that performs a specific task when executed by a computer.

From the program in its human-readable form of code, a compiler (a program that converts instructions into a machine-code) can derive machine code—a form consisting of instructions that the computer can directly execute.

### **1. Program structure:**

Programs are frequently referred to as either well-structured or poorly structured. A well-structured program employs appropriate data structures and program units with a single entry point and a single exit point, while a poorly structured program has arbitrary data structures and flow of control.

The overall pattern is:

- Statements to establish the start of the program
- Variable declaration
- Program statements

### **2. Variable declaration:**

Variables are the names you give to computer memory locations which are used to store values in a computer program.

For example, assume you want to store two values 10 and 20 in your program and at a later stage, you want to use these two values. Let's see how you will do it. Here are the following three simple steps –

- Create variables with appropriate names.
- Store your values in those two variables.
- Retrieve and use the stored values from the variables.

Ex: `int count;`

where, *int* is Type; *Count* is Name

Variables are the backbone of any program, and thus the backbone of any programming language.

### **3. Data type:**

Unlike humans, a computer does not know the difference between "1234" and "abcd." A data type is a classification of the type of data that a variable or object (Units of code that are eventually derived from the process) can hold in computer programming.

Different types of variables include:

- Integer(int) – to store integer or “whole” numbers
- Real – to store real or fractional numbers (also called float to indicate a floating point number)
- Character – A single character such as a letter of the alphabet or punctuation.
- String – A collection of characters
- Boolean – True or False
- Double (e.g., 1.79769313486232E308)
- Date (e.g., 03/01/2016)

In order to use a variable within a program, the compiler needs to know in advance the type of data that will be stored in it. For this reason, we declare the variables at the start of the program. Variable declaration consists of giving a new name and a data type for the variable. This is normally done at the very start of the program.

### **4. Looping structures:**

Loop structures allow you to run one or more lines of code repetitively. You can repeat the statements in a loop structure until a condition is True, until a condition is False, a specified number of times, or once for each element in a collection.

The following illustration shows a loop structure that runs a set of statements until a condition becomes true.

### **5. Control structures:**

A control structure is a block of programming that analyses variables and chooses a direction in which to go based on given parameters.

When a program is running, the code is being read by the computer line by line (from top to bottom, and for the most part left to right), just like you would read a book. This is known as the “code flow “, now as the code is being read from top to bottom, it may hit a point where it needs to make a decision, this decision could make the code jump to a completely different part of the program, or it could make it re-run a certain piece again, or just plain skip a bunch of code. The

computer program has a strict set of rules to decide which direction to go. So, this decision that must be made, that will in turn affect the flow of code, is known as a control structure.

## **6. Syntax:**

In computer science, the syntax of a programming language is the set of rules that define the combinations of symbols that are considered to be correctly structured programs in that language. Documents that are syntactically invalid are said to have a syntax error.

The concept of syntax in programming language is similar to the concepts of grammar and spelling in spoken language. When a sentence in English has very poor grammar and spelling, it becomes difficult or even impossible to understand sometimes. Similarly, when code has syntax errors, the program will not execute.

The difference is that when you read a sentence in English with a minor error, you can typically still understand its meaning. Very small syntax errors in code, however, will not allow the program to run. Coders, therefore, have to pay great attention to detail to make sure their code is not only logical but also free of syntax errors. In today's world, where information is just a 'search' away, syntax is very easy to learn

## **6. Data structure:**

A data structure is a specialized format for organizing and storing data.

In computer programming, a data structure may be selected or designed to store data for the purpose of working on it with various algorithms.

## **7. Algorithms:**

Algorithm is simply a sequence of steps to solve a problem. These steps can be any combination of logical operators (and/or), mathematical operators (add, subtract, count etc.), text operators (combine two words, extract first 3 alphabets from a word etc.) conditional statements (if/then/else) and many others depending on the complexity of the problem at hand. Algorithm is basically like a recipe when you prepare a dish – a sequence of operations to achieve an output (dish)

Data structures are used to hold data while algorithms are used to solve the problem using that data. It also teaches you the science of evaluating the efficiency of an algorithm. This enables you to choose the best of various choices.

## **8. Function:**

A function is a block of code that performs a specific task. Function takes a set of inputs called arguments and provides an output as per the definition of the function. A function contains a unit

of code that is designed to perform a specific task – which could be an algorithm. For example, you could define a function  $F$  which takes two arguments  $a$  &  $b$  and performs addition on those two. So, you could run a code that looks like  $F(2,3)$  and get 5 as an output. Now this was a very simple example for illustration purpose; in real life problem solving, a function could be of varied complexities; e.g. a function called  $\text{plot}(x,y)$  can create a chart for you with the values stores in  $x$  &  $y$  variables.