

## Flow of control Looping statement

It is also called a repetitive control structure. Sometimes we require a set of statements to be executed many times by changing the value of one or more variables each time to obtain a different result. This type of program execution is called looping. C++ provides the following construct

- while loop
- do-while loop
- for loop

### While loop

Syntax of while loop while(condition)

```
{  
statement(s);  
}
```

The flow diagram indicates that a condition is first evaluated. If the condition is true, the loop body is executed and the condition is re-evaluated. Hence, the loop body is executed repeatedly if the condition remains true. As soon as the condition becomes false, it comes out of the loop and goes to the state next to the 'while' loop.

### do-while loop

Syntax of do-while loop do

```
{ statements;  
} while (condition);
```

Note: That the loop body is always executed at least once. One important difference between the while loop and the do-while loop is the relative ordering of the conditional test and loop body execution. In the while loop, the loop repetition test is performed before each execution of the loop body; the loop body is not executed at all if the initial test fails. In the do-while loop, the loop termination test is performed after each execution of the loop body. Hence, the loop body is always executed at least once.

## for loop

It is a count controlled loop in the sense that the program knows in advance how many times the loop is to be executed.

syntax of for loop for (initialization; decision; increment/decrement)

```
{  
statement(s);  
}
```

In the For loop Three operation takes place

- Initialization of loop control variable
- Testing of loop control variable
- Update the loop control variable either by incrementing or decrementing.

Operation (i) is used to initialize the value. On the other hand, operation (ii) is used to test whether the condition is true or false. If the condition is true, the program executes the body of the loop and then the value of loop control variable is updated. Again, it checks the condition and so on. If the condition is false, it gets out of the loop

### Jump Statements

The jump statements unconditionally transfer program control within a function.

- goto statement
- break statement

- continue statement

## The goto statement

goto allows making the jump to another point in the program. goto pqr;

pqr: pqr is known as a label. It is a user-defined identifier. After the execution of goto statement, the control transfers to the line after label pqr.

## The break statements

The break statement, when executed in a switch structure, provides an immediate exit from the switch structure. Similarly, you can use the break statement in any of the loops. When the break statement executes in a loop, it immediately exits from the loop.

## ARRAY

An array is a collection of data elements of same data type. It is described by a single name and each element of an array is referenced by using array name and its subscript no.

### Declaration of Array

Type arrayname[numberofelements]; For example, int Age[5] ;  
float cost[30];

### Initialization of One-Dimensional Array

An array can be initialized along with declaration. For array initialization, it is required to place the elements separated by commas enclosed within braces. int A[5] = {11,2,23,4,15}; It is possible to leave the array size open. The compiler will count the array size. int B[] = {6,7,8,9,15,12};

### Referring to Array Elements

In any point of a program in which an array is visible, we can access the value of any of its elements individually as if it was a normal variable, thus being able to both read and modify its value. The format is as simple as:

name[index]

Examples: cout<<age[4]; //print an array element age[4]=55;

Using LooptoinputanArrayfromuser int age [10], i ;

for (i=0 ; i<10; i++)

```
{  
cin>>age[i];  
}
```

### Arrays as Parameters

At some moment, we may need to pass an array to a function as a parameter. In C++, it is not possible to pass a complete block of memory by value as a parameter to a function, but we can pass its address.

Forexample, the following function: void print(int A[]) accepts a parameter of type "array of int" called A.

In order to pass to this function, an array declared as int arr[20]; we need to write a call like this: print(arr);