**Lecture 7**

**Loops**

* There may be a situation, when you need to execute a block of code several number of times. In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on.
* Programming languages provide various control structures that allow for more complicated execution paths.
* A loop statement allows us to execute a statement or group of statements multiple times and following is the general from of a loop statement in most of the programming languages

**Loop Types:**

C++ programming language provides the following type of loops to handle

looping requirements.

|  |  |
| --- | --- |
| **Loop Type** | **Descryption** |
| **for loop** | The initialization, condition and increment/decrement all put together. Initialization will be done once at the beginning of loop. Then, the condition is checked by the compiler. If the condition is false, for loop is terminated. But, if condition is true then, the statements are executed until condition is false. |
| **while Loop** | Repeats a statement or group of statements while given condition is true. It tests the condition before executing the loop body. |
| **do...while loop** | Like a ‘while’ statement, except that it tests the condition at the end of the loop body. |
| **nested loops** | You can use one or more loop inside any another ‘while’, ‘for’ or ‘do..while’ loop. |

**For Loop:** A **for** loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

**Syntax of For Loop**

 for ( initialization ; condition; increment )

 {

 Statements ;

 }

**Here is the flow of control in a for loop:**

* The ( initializationstep is executed first, and only once. This step allows you to declare and initialize any loop control variables. You are not required to put a statement here, as long as a semicolon appears.
* Next, the **condition** is evaluated. If it is true, the body of the loop is executed. If it is false, the body of the loop does not execute and flow of control jumps to the next statement just after the for loop.
* After the body of the for loop executes, the flow of control jumps back up to the **increment** statement. This statement allows you to update anyloop control variables. This statement can be left blank, as long as a semicolon appears after the condition.
* After the condition becomes false, the for loop terminates.

**Example:**

#include<iostream.h>

 void main()

 {

 int a, num;

 cout << "Enter any number : ";

 cin >> num;

 for (a=1;a<=num;a++)

 cout << "\nHello...!!"; }

**The output:**

 Enter any number : 5

 Hello...!!

 Hello...!!

 Hello...!!

 Hello...!!

 Hello...!!

**While Loop**

While loop is also called entry control loop because, in while loop, compiler will 1st check the condition, whether it is true or false, if condition is true then execute the statements.

**Syntax of While Loop**

 while ( condition )

 {

 Statmentes;

 --- - -- - -- -- - -

 }

* Here, **statement(s)** may be a single statement or a block of statements.
* The **condition** may be any expression, and true is any non-zero value. The loop iterates while the condition is true. When the condition becomes false, program control passes to the line immediately following the loop.

**Example**

#include <iostream.h>

//using namespace std;

int main ()

{

// Local variable declaration:

int a = 10;

// while loop execution

while( a < 20 )

{

cout << "value of a: " << a << endl;

a++;

}return o;}

**The output:**

value of a: 10

value of a: 11

value of a: 12

value of a: 13

.

.

value of a: 19

 **يقوم هذا البرنامج بطباعة قيمة المتغير a طالما الشرط (( a<20 متحقق او TRUE**

**Homework: trace the following c++ cods and Find the final output for these cods :**

#include <iostream.h>

int main()

{

 int n, sum = 0;

 cout << "Enter a positive integer: ";

 cin >> n;

 for (int i = 1; i <= n; ++i) {

 sum += i;

 }

 cout << "Sum = " << sum;

}

#include <iostream.h>

int main()

{

int n = 10 ;

 while (n > 0 (

{

 cout << n <<" ," ;

 -- n ;

 }

 cout << "End of program \n";

 return 0;

}

**Exercises**

1. Write C++ program to find the summation of the following series: Sum= 1+3+5+7+ . . . . . . . . +99

****

1. Write C++ program to read 10 integer numbers, and find the sum of the positive numbers only.

****

1. **Write C++ program to find the summation of the following series :**

** **

1. **Write C++ program to find the summation of students marks, and its average, assume the student have 8 marks.**

****

**Lecture 8**