National University of Computer and Emerging Sciences Islamabad Programming Fundamentals Lab FALL 2022

### Lab Manual 10

**Repetitions-I (while loop, do…while loop, for loop)**

**In computer programming, loops are used to repeat a block of code.**

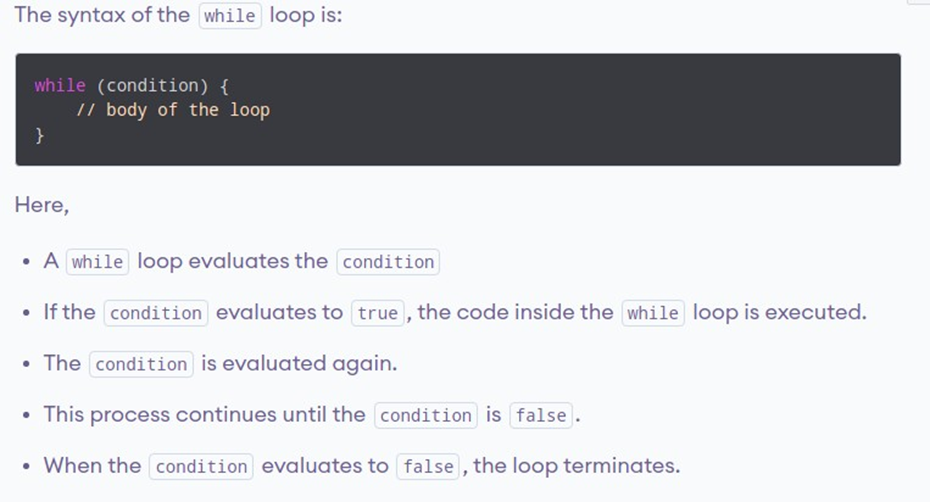
A loop is used for executing a block of statements repeatedly until a particular condition is satisfied. For example, when you are displaying number from 1 to 100 you may want set the value of a variable to 1 and display it 100 times, increasing its value by 1 on each loop iteration instead of writing the print statement 100 times.

There are 3 types of loops in C++.

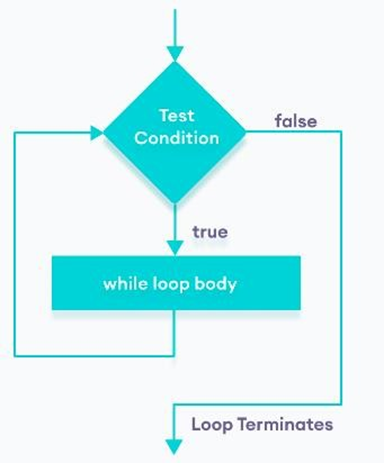
1. for loop
2. while loop
3. do...while loop

# While loop:

In while loop, condition is evaluated first and if it returns true then the statements inside while loop execute, this happens repeatedly until the condition returns false. When condition returns false, the control comes out of loop and jumps to the next statement in the program after while loop.

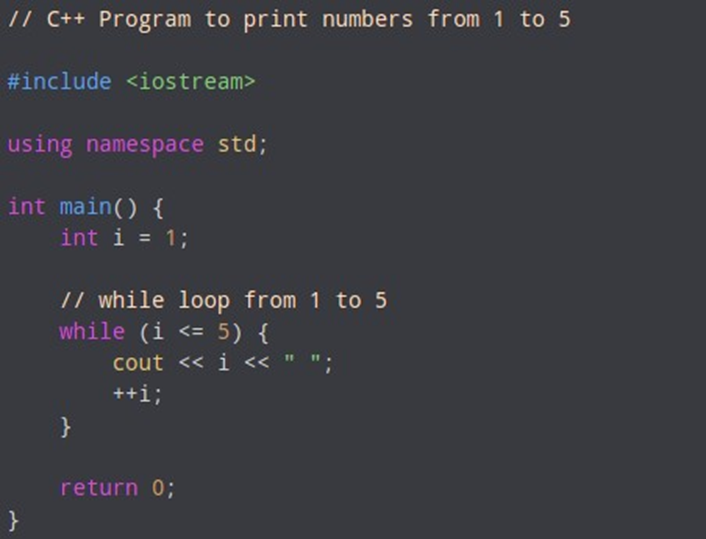


### Flowchart of while loop:

****

**Example 1.1:**

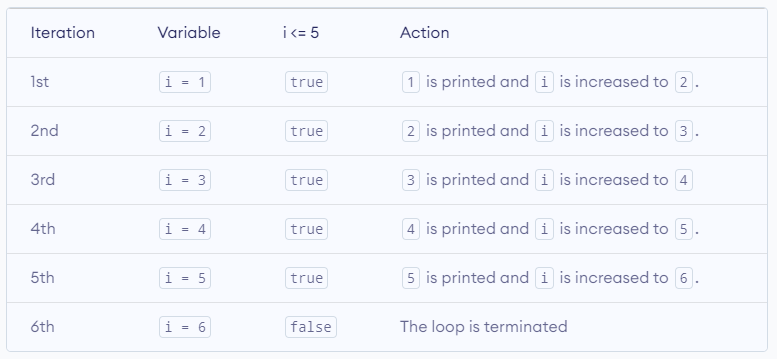
Display numbers from 1 to 5.



### Output:

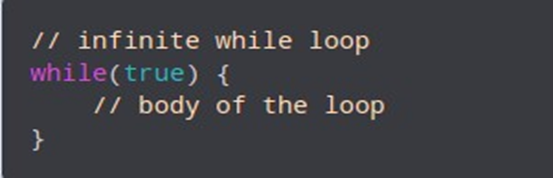
**1 2 3 4 5**

**Working of program:**



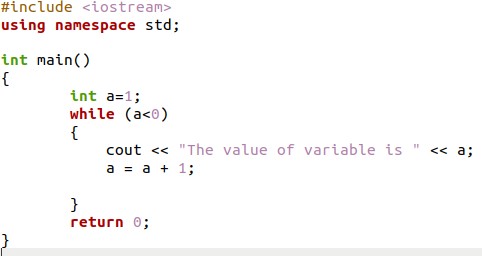
**Infinite while loop:**

A while loop that never stops is said to be the infinite while loop, when we give the condition in such a way so that it never returns false, then the loops becomes infinite and repeats itself indefinitely.



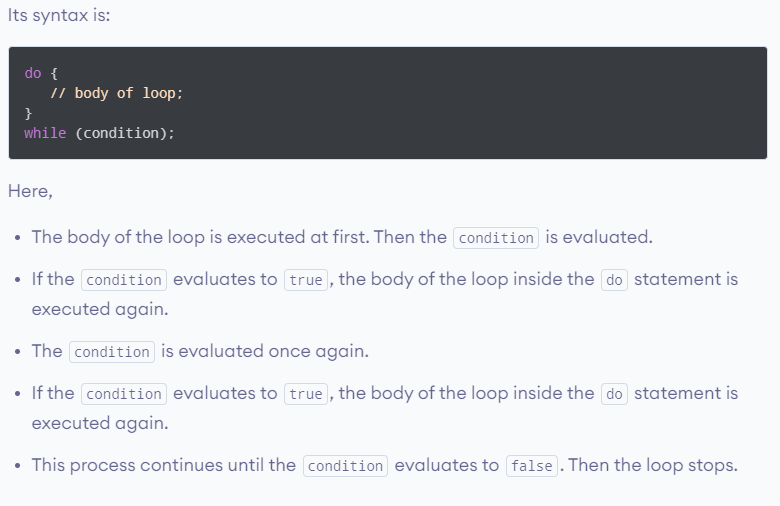
### Example 1.2:

3

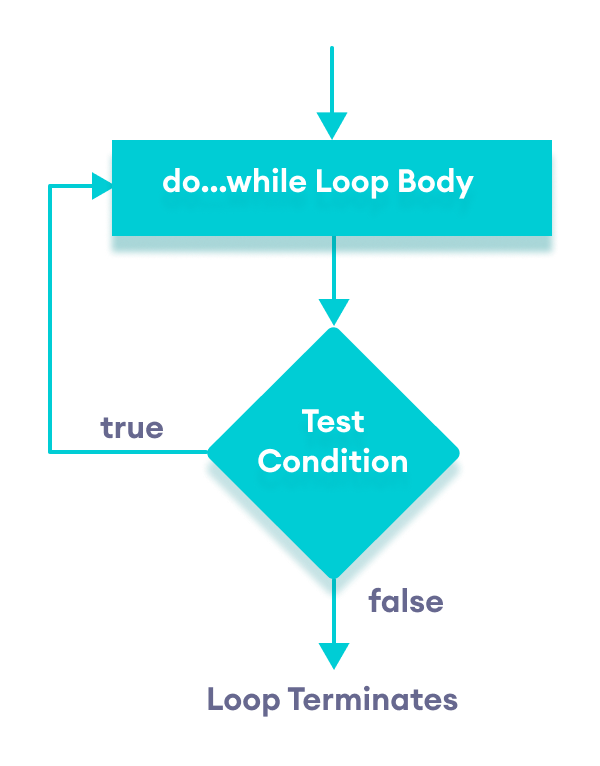


# Do…While loop:

The do...while loop is a variant of the while loop with one important difference: the body of do...while loop is executed once before the condition is checked.

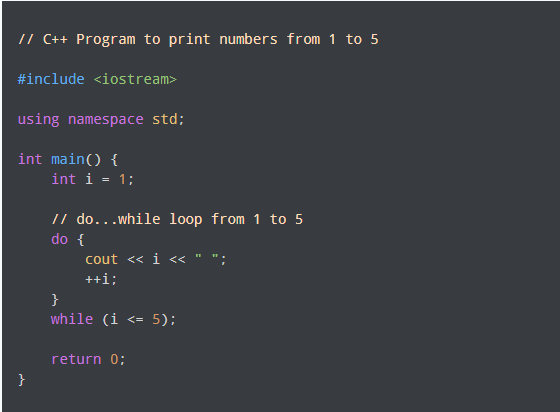


### Flowchart of do…while loop:



**Example 1.1:**

Display numbers from 1 to 5.



### Output:

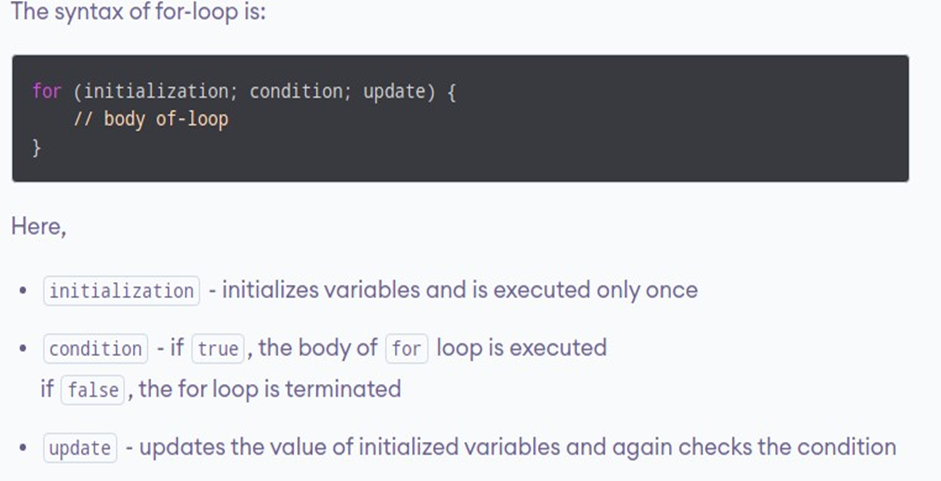
**1 2 3 4 5**

**Working of program:**

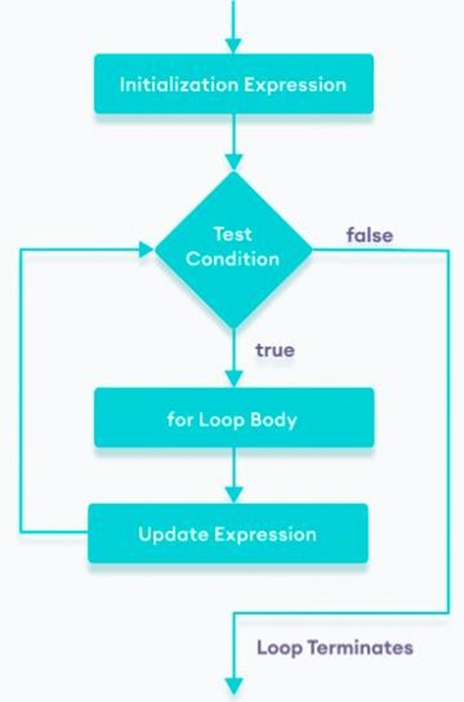


**For Loop:**

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.

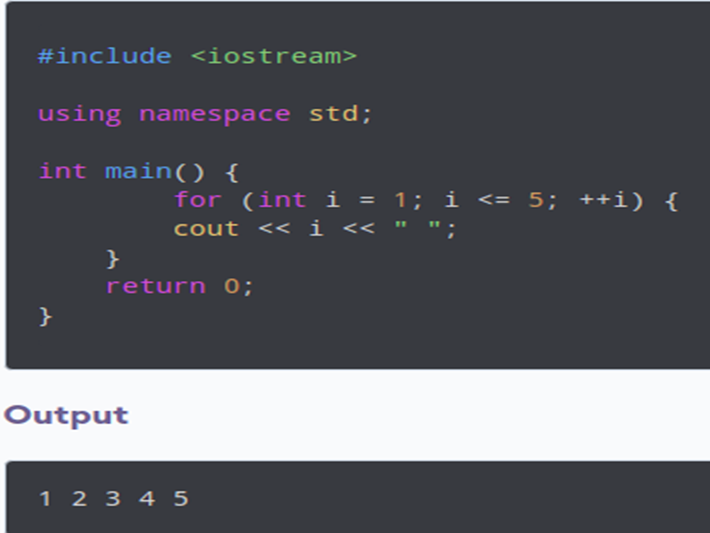


### Flowchart of for loop:



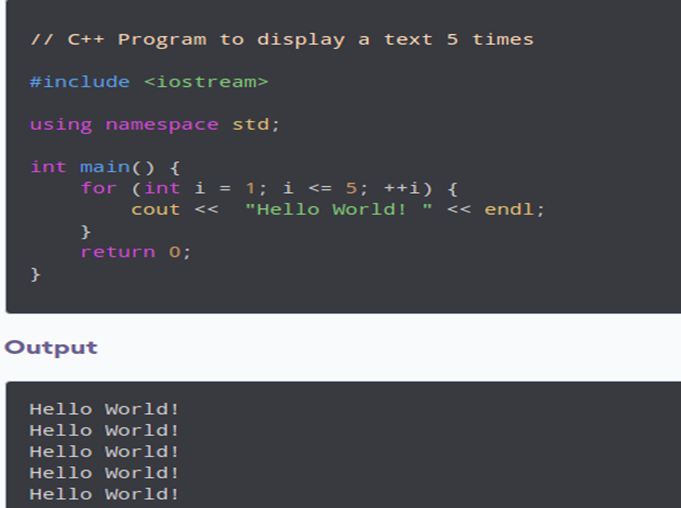
**Example 2.1:**

Display numbers from 1 to 5



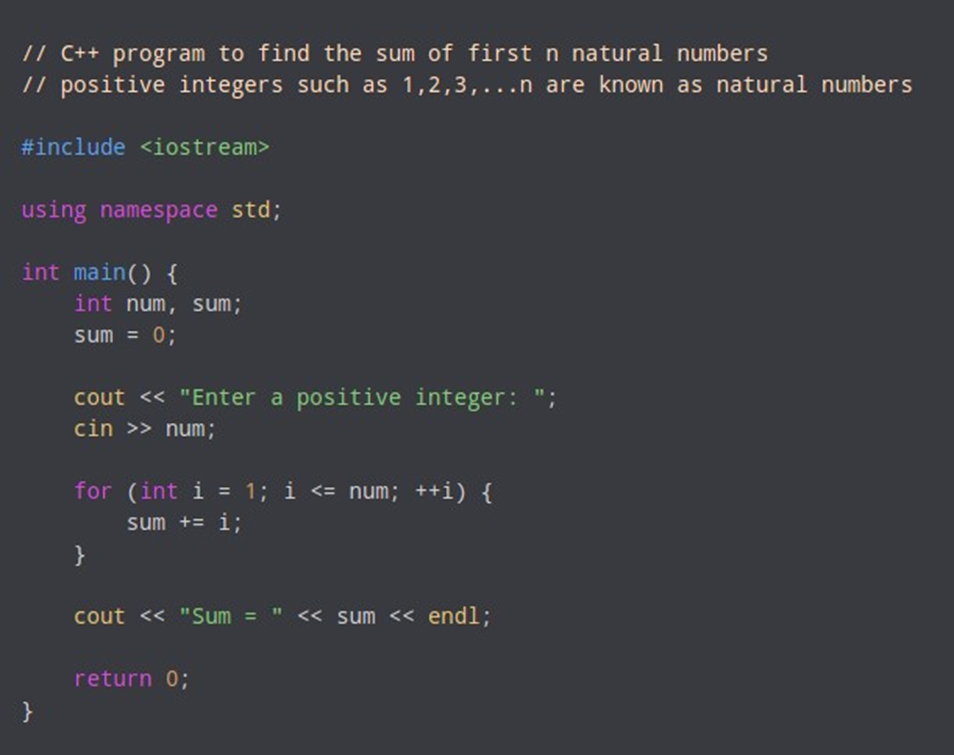
**Example 2.2:**

Display “Hello world” 5 times.



## Example 2.3:

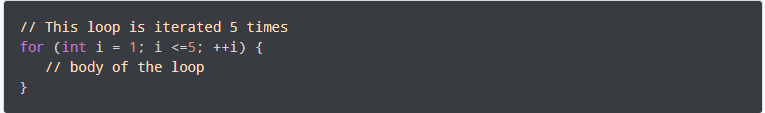
Find the sum of first n Natural Numbers.





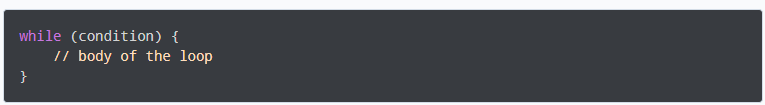
**For Loop VS While or Do…While Loop**

A for loop is usually used when the number of iterations is known. For example,



Here, we know that the for-loop will be executed 5 times.

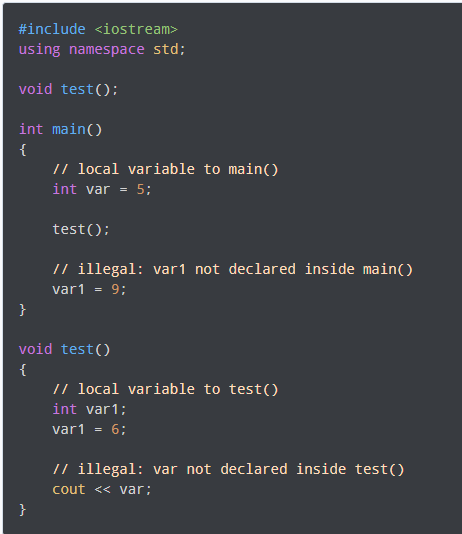
However, while and do...while loops are usually used when the number of iterations is unknown. For example,



**Local Variable, Global Variable and Static Variable**

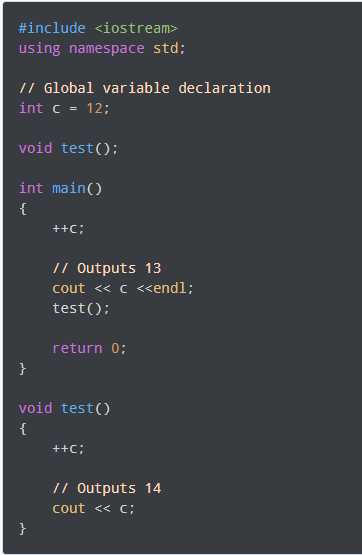
**Local Variable**

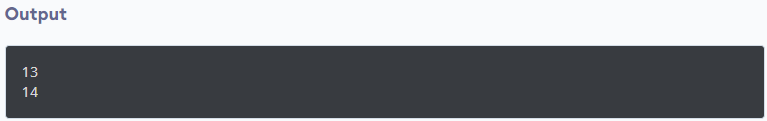
A variable defined inside a function (defined inside function body between braces) is called a local variable or automatic variable. Its scope is only limited to the function where it is defined. In simple terms, local variable exists and can be accessed only inside a function. The life of a local variable ends (It is destroyed) when the function exits.



**Global Variable**

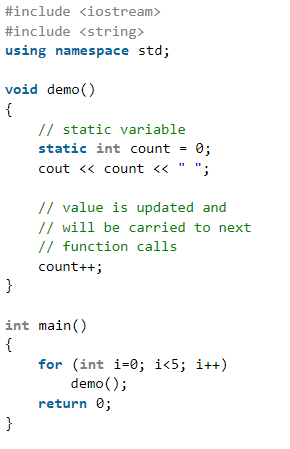
If a variable is defined outside all functions, then it is called a global variable. The scope of a global variable is the whole program. This means, it can be used and changed at any part of the program after its declaration. Likewise, its life ends only when the program ends.

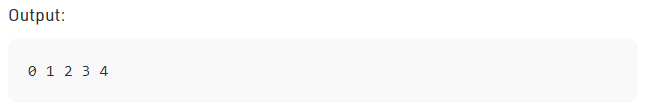




**Static Variable**

When a variable is declared as static, space for it gets allocated for the lifetime of the program. Even if the function is called multiple times, space for the static variable is allocated only once and the value of variable in the previous call gets carried through the next function call.





# Lab Tasks

# Part A

You have been provided with a text file. You have to identify and describe errors from each part then fix that error.

# Part B

Problem solving using C++.

# Note: You have to make a Function in each task.

# Problem 01:

Write a program that calculates the balance of a savings account at the end of a period of time. It should ask the user for the annual interest rate, the starting balance, and the number of months that have passed since the account was established. A loop should then iterate once for every month, performing the following:

1. Ask the user for the amount deposited into the account during the month. (Do not accept negative numbers.) This amount should be added to the balance.
2. Ask the user for the amount withdrawn from the account during the month. (Do not accept negative numbers.) This amount should be subtracted from the balance.
3. Calculate the monthly interest. The monthly interest rate is the annual interest rate divided by twelve. Multiply the monthly interest rate by the balance, and add the result to the balance.

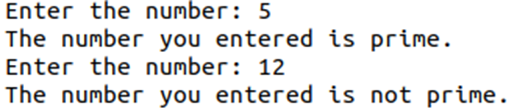
After the last iteration, the program should display the ending balance, the total amount of deposits, the total amount of withdrawals, and the total interest earned.

**Note:** If a negative balance is calculated at any point, a message should be displayed indicating the account has been closed and the loop should terminate.

# Problem 02:

Create a C++ program which takes an integer number as input and pass it thorough parameter to the function named ***checkPrimeNumber***. This function ***checkPrimeNumber*** should display whether the number is prime or not. This process should continue until the user enter negative number.

**Expected Output:**

****

# Problem 03:

Write a program that asks the user to enter today’s sales for five stores. The program should then display a bar graph comparing each store’s sales. Create each bar in the bar graph by displaying a row of asterisks. Each cross(x) should represent $50 of sales.

Here is an example of the program’s output.

Enter today's sales for store 1: 1000

Enter today's sales for store 2: 1200

Enter today's sales for store 3: 1800

Enter today's sales for store 4: 800

Enter today's sales for store 5: 2000

SALES BAR CHART

(Each x = $50)

Store 1: **xxxxxxxxxxxxxxxxxxxx**

Store 2: **xxxxxxxxxxxxxxxxxxxxxxxx**

Store 3: **xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx**

Store 4: **xxxxxxxxxxxxxxxx**

Store 5**: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx**

|  |
| --- |
| **Submission Instructions:**   1. Save all .cpp files and screenshot with Question number e.g. Q1.cpp 2. Now create a new folder with name NAME\_ROLLNO\_LAB04 e.g. XYZ\_i22XXXX\_LAB10 3. Move all of your .cpp files to this newly created directory and compress it into .zip file. 4. Now you have to submit this zipped file on Google Classroom. |