



CL1002-Programming Fundamentals— FALL 2023

LAB 07

Loops



Learning Outcomes

In this lab you are expected to learn the following:

- For loops
- While loops



Loops:

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 the while loop execute, this happens repeatedly until the condition returns false. When the condition returns false, the control comes out of loop and jumps to the next statement in the program after while loop.

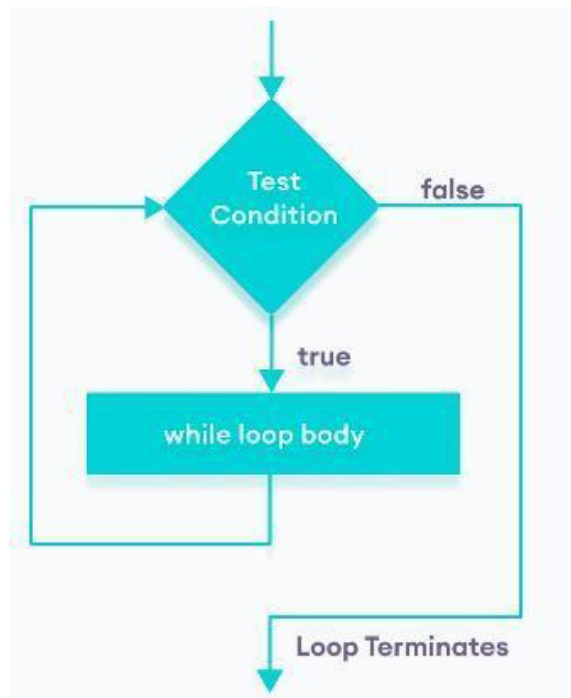
The syntax of the `while` loop is:

```
while (condition) {  
    // body of the loop  
}
```

Here,

- A `while` loop evaluates the `condition`
- If the `condition` evaluates to `true`, the code inside the `while` loop is executed.
- The `condition` is evaluated again.
- This process continues until the `condition` is `false`.
- When the `condition` evaluates to `false`, the loop terminates.

Flowchart of while loop:



Example 1.1:

Display numbers from 1 to 5.

```
// C++ Program to print numbers from 1 to 5

#include <iostream>

using namespace std;

int main() {
    int i = 1;

    // while loop from 1 to 5
    while (i <= 5) {
        cout << i << " ";
        ++i;
    }

    return 0;
}
```

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.

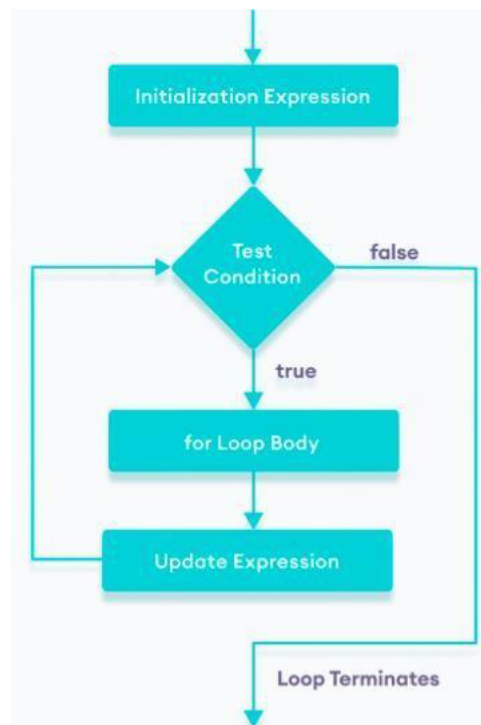
The syntax of for-loop is:

```
for (initialization; condition; update) {  
    // body of-loop  
}
```

Here,

- `initialization` - initializes variables and is executed only once
- `condition` - if `true`, the body of `for` loop is executed
if `false`, the for loop is terminated
- `update` - updates the value of initialized variables and again checks the condition

Flowchart of For loop:



Example 2.1:

Display numbers from 1 to 5



```
#include <iostream>

using namespace std;

int main() {
    for (int i = 1; i <= 5; ++i) {
        cout << i << " ";
    }
    return 0;
}
```

Submission Instructions:

1. Save the **cpp** file with the roll no and task number
e.g. i230001_Q1.cpp
2. Now create a new folder with *ROLLNO_LAB01_SEC* e.g. i23XXXX_LAB08_A
3. You need to display your roll no and name before the output of each question.
4. Now you have to submit this zipped file on Google Classroom.
5. If you don't follow the above-mentioned submission instructions, you will be marked **zero**.
6. Plagiarism in the Lab Task will result in **zero** marks in the whole category.

Lab Tasks

Solve Tasks 1 to 3 Using For Loop:

Task 01

In mathematics, the factorial of a positive integer n , denoted by $n!$, is the product of all positive integers less than or equal to n . For Example $5! = 5 * 4 * 3 * 2 * 1$. Write a program using loops that can compute the factorial of an integer entered by the user. If the number is less than zero display "invalid Input" else display the resulting factorial.

Note: $0! = 1$

Output:

```
Enter a positive integer: 5
Factorial of 5 = 120
```

Task 02

Write a program that calculates the occupancy rate for a hotel. The program should start by asking the user how many floors the hotel has. A loop should then iterate once for each floor. In each iteration, the loop should ask the user for the number of rooms on the floor and how many of them are occupied. After all the iterations, the program should display how many



rooms the hotel has, how many of them are occupied, how many are unoccupied, and the percentage of rooms that are occupied. The percentage may be calculated by dividing the number of rooms occupied by the number of rooms.

Output:

```
Enter number of floors: 3
Enter the number of rooms on the floor 1 11
How many rooms are occupied? 8
Enter the number of rooms on the floor 2 12
How many rooms are occupied? 7
Enter the number of rooms on the floor 3 13
How many rooms are occupied? 9
The hotel has total of 3 floors
The hotel has total of 36 rooms
There are 24 rooms occupied
There are 12 empty rooms
Percentage of occupied rooms is 66.6667%
```

Task 03

Write a program that prints the Fibonacci Series. The program will take a number as input and print the Fibonacci series up to that number. In the Fibonacci series, the next number is the sum of previous two numbers for example 0, 1, 1, 2, 3, 5, 8, 13, 21 etc. The first two numbers of Fibonacci series are 0 and 1.

Output:

```
Enter the number of elements: 10
0 1 1 2 3 5 8 13 21 34
```

Solve Tasks 4 and 5 Using While Loop:

Task 04

Write a program that designs a guessing game using loops. The program will think of a random number between 1 and 100. The user will be given 10 attempts to guess the number. If the guessed number is greater than the actual number your program should print too high. If the number is less than the actual number then your program should print too low. If the user is unable to guess the number in 10 attempts, you should print that you ran out of tries. If the user guesses correctly within the 10 attempts, your program should terminate and display that you guessed correctly.

The sample output is shown below:



Output:

```
Hi! I'm thinking of a random number between 1 and 100.
--- Attempt 1
Guess what number I am thinking of: 50
Too high.
--- Attempt 2
Guess what number I am thinking of: 25
Too high.
--- Attempt 3
Guess what number I am thinking of: 17
Too high.
--- Attempt 4
Guess what number I am thinking of: 9
Too low.
--- Attempt 5
Guess what number I am thinking of: 14
Too high.
--- Attempt 6
Guess what number I am thinking of: 12
Too high.
...
--- Attempt 10
Guess what number I am thinking of: 10
Too low.

Aw, you ran out of tries. The number was 11.
```

Task 05

Write a C++ program which takes an integer number as input and checks whether the number entered by the user is a prime number. An integer is said to be a prime number if and only if n is greater than 1 and is divisible only by 1 and n . For example, 17 and 29 are prime numbers, whereas, 1 and 28 are not prime numbers. This process should continue until the user enters a negative number.

Output:

```
Enter the number: 5
The number you entered is prime.
Enter the number: 12
The number you entered is not prime.
Enter the number: -9
```