National University
of Computer & Emerging Sciences

CL1002-Programming Fundamentals– FALL 2023

LAB 10

Loops + Functions

Learning Outcomes

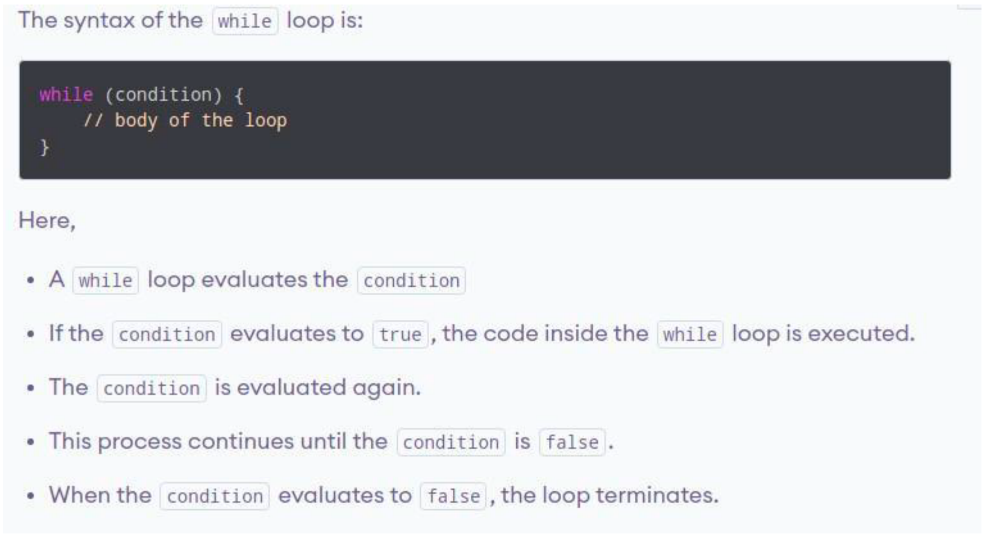
In this lab you are expected to learn the following:

- Nested Loops (Revision)
- Functions

Loops:

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.



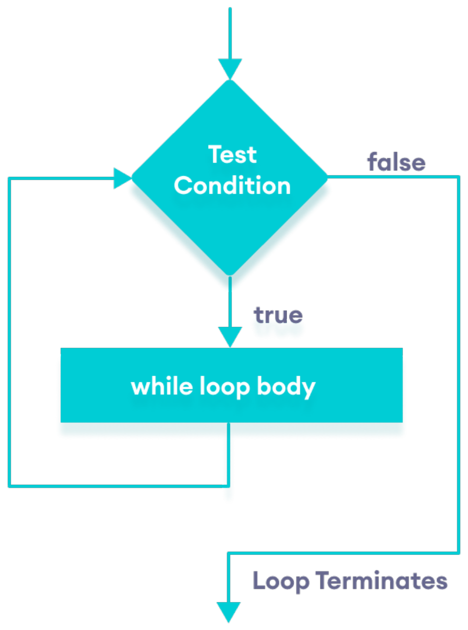
The syntax of the `while` loop is:

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

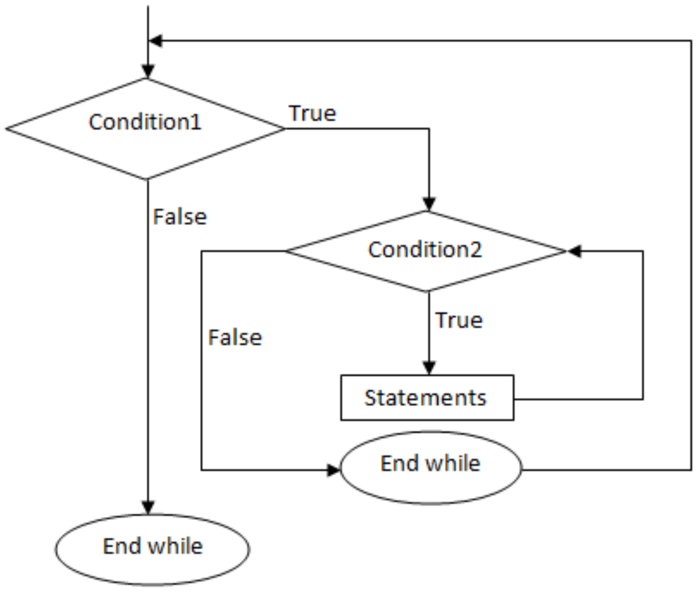
Here,

- 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:

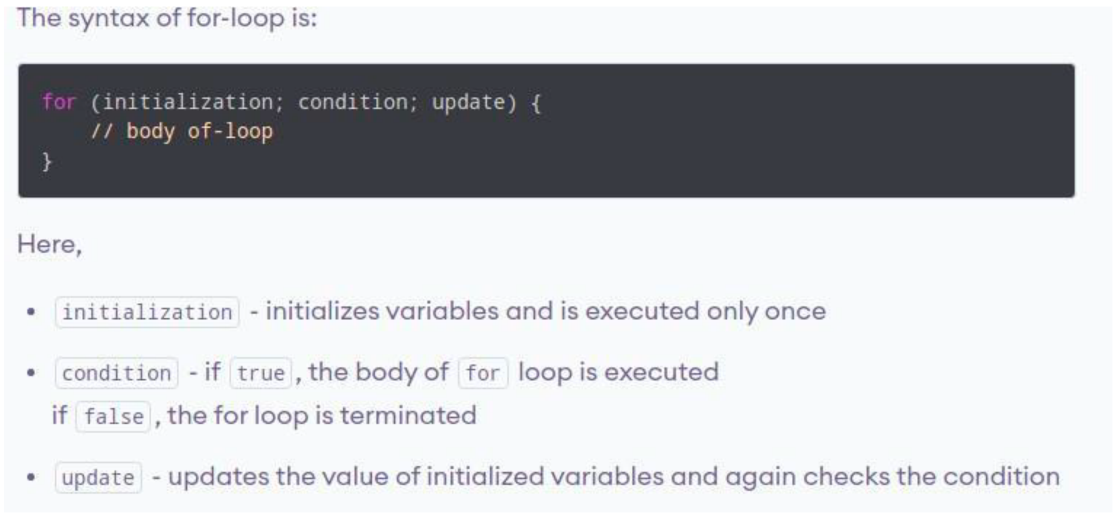


Flow chart for nested while loop



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.



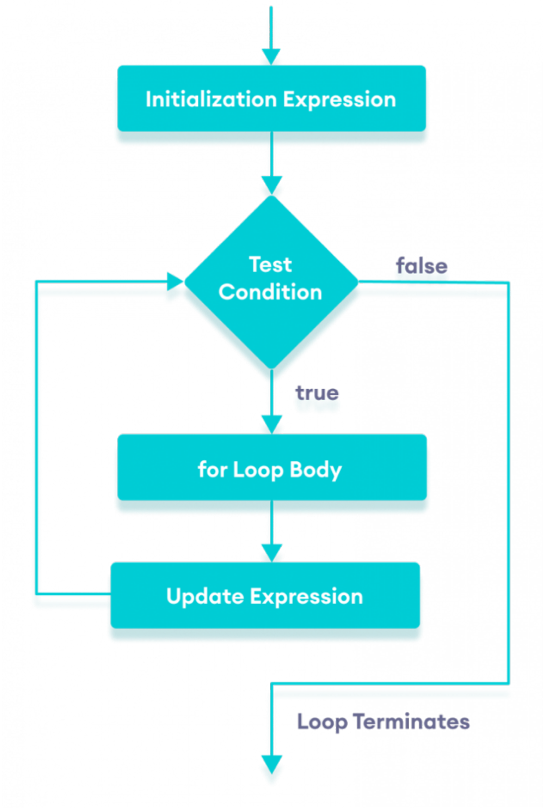
The syntax of for-loop is:

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

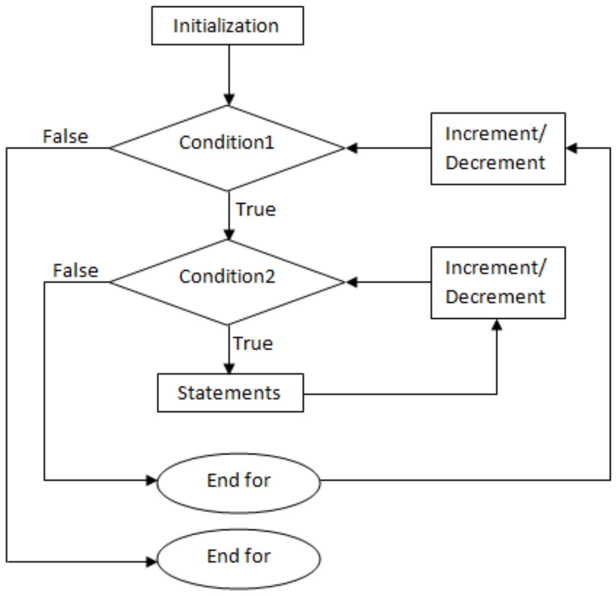
Here,

- `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:



Flowchart of nested for loop



Functions:

A function is a block of code that performs a specific task. Functions are commonly used to break a problem down into small manageable pieces. Instead of writing one long function that contains all of the statements necessary to solve a problem, several small functions that each solve a specific part of the problem can be written.

There are two types of function:

1. **Standard Library Functions:** Predefined in C++
2. **User-defined Function:** Created by users

C++ User-defined Function

C++ allows the programmer to define their own function. A user-defined function groups code to perform a specific task and that group of code is given a name (identifier). When the function is invoked from any part of the program, it all executes the codes defined in the body of the function.

Function Declaration

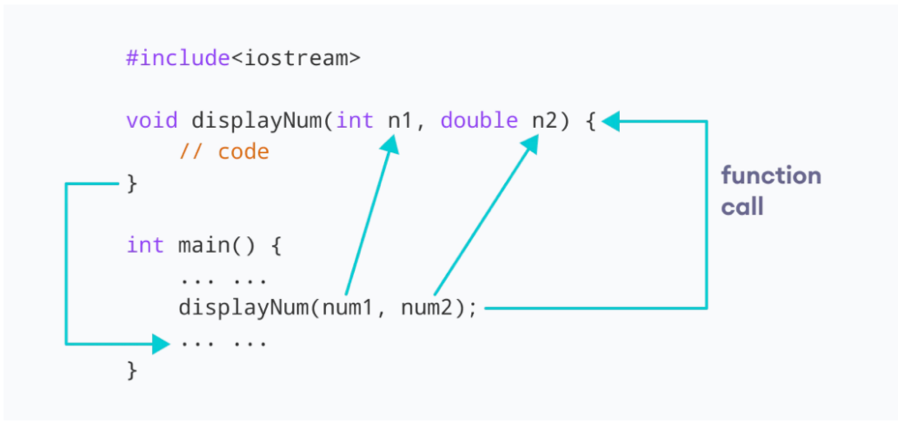
The function declaration consists of the return type, the name of the function, and parameters (if any). The syntax to declare a function is:

```
returnType functionName (parameter1, parameter2,...) {
    // function body
}
```

Function Definition:

A function definition contains the statements that make up the function. When creating a function, you must write its definition. All function definitions have the following parts:

1. **Return type:** A function can send a value to the part of the program that executed it. The return type is the data type of the value that is sent from the function.
2. **Name:** You should give each function a descriptive name. In general, the same rules that apply to variable names also apply to function names.
3. **Parameter list:** The program can send data into a function. The parameter list is a list of variables that hold the values being passed to the function.
4. **Body:** The body of a function is the set of statements that perform the function’s operation. They are enclosed in a set of braces.



```
#include<iostream>

void displayNum(int n1, double n2) {
    // code
}

int main() {
    ... ...
    displayNum(num1, num2);
    ... ...
}
```

In the above example, we have used a function that has one int parameter and one double parameter. We then pass num1 and num2 as arguments. These values are stored by the function parameters n1 and n2 respectively.

Function Calling

Declared functions are not executed immediately. They are "saved for later use", and will be executed later, when they are called. To call a function, write the function's name followed by two parentheses () and a semicolon;. The syntax to call a function is:

```
functionName (parameter1, parameter2,...)
```

Return Statement

The return statement causes a function to end immediately. When the last statement in a function has finished executing, the function terminates and the program returns to the statement following the function call. It's possible, however, to force a function to return before the last statement has been executed. When the return statement is encountered, the function immediately terminates and control of the program returns to the statement that called the function.

Forward Declaration of a Function:

A forward declaration allows us to tell the compiler about the existence of an identifier before actually defining the identifier. In the case of functions, this allows us to tell the compiler about the existence of a function before we define the function's body.

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_LAB10_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

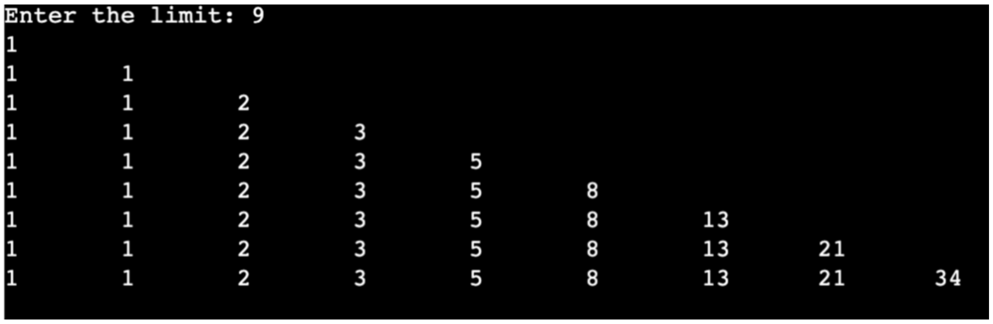
Task 01

The Fibonacci numbers are the numbers in the following integer series.

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

Write a program that prints Fibonacci Triangle using nested for loops as shown. The program should take the limit of Fibonacci series and print the triangle accordingly.

Output:

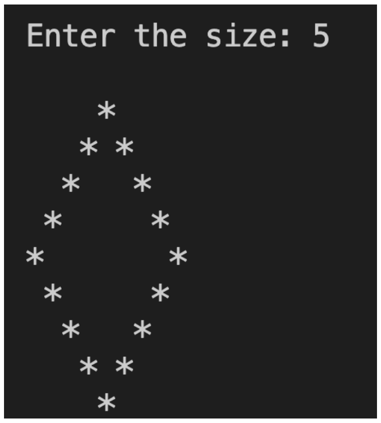


```
Enter the limit: 9
1
1       1
1       1       2
1       1       2       3
1       1       2       3       5
1       1       2       3       5       8
1       1       2       3       5       8       13
1       1       2       3       5       8       13      21
1       1       2       3       5       8       13      21      34
```

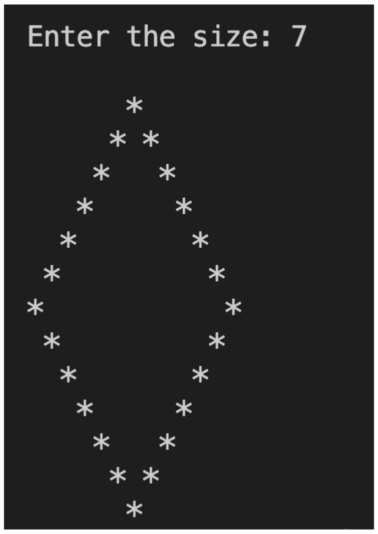
Task 02

Write a C++ program to display the hollow diamond pattern using nested loops. The program will ask the user to input the number of rows and display the hollow diamond pattern accordingly.

Output:



```
Enter the size: 5

    *
   * *
  *   *
 *     *
*       *
 *     *
  *   *
   * *
    *
```



```
Enter the size: 7

      *
     * *
    *   *
   *     *
  *       *
 *         *
*           *
 *         *
  *       *
   *     *
    *   *
     * *
      *
```

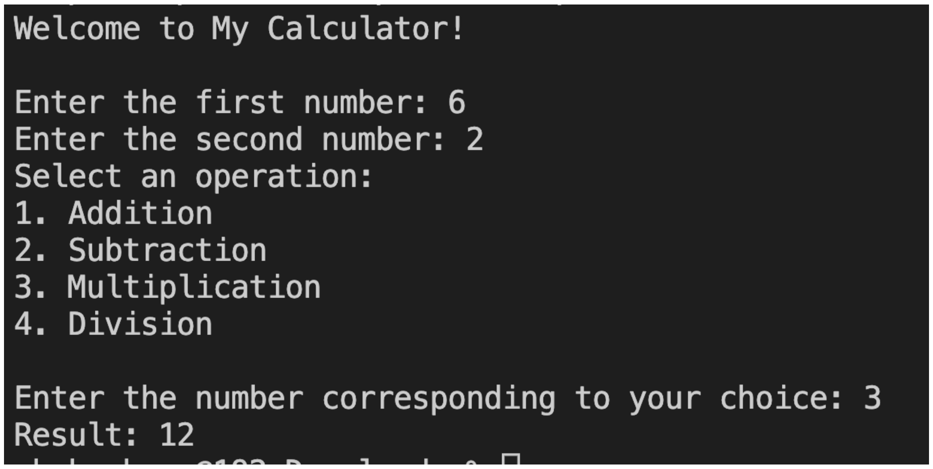
Task 03

Write a program that will play a game with the user, called Odds and Evens. The computer will play Evens, and the human user will play Odds. For a round of the game, each player picks an integer in the range [1,10]. The players pick their numbers independently: neither player knows the other player's number before choosing its own number. The computer will pick up a random number using rand() function whereas the user will input the number. If the sum of the numbers is even, then Evens (the computer) wins that round; if the sum of the numbers is odd, then Odds (the human) wins that round. The game continues for as many rounds as the user wants to play; the user ends the game by typing a non-number or a number outside [1,10] for the input. At the end of the game, the program summarizes the score.

Task 04

Write a program to implement the functionality of a calculator using functions. The calculator should be able to perform addition, subtraction, multiplication, and division operations. Each operation should be implemented as a separate function. Prompt the user to enter two numbers and select an operation and then display the result of the chosen operation.

Output:



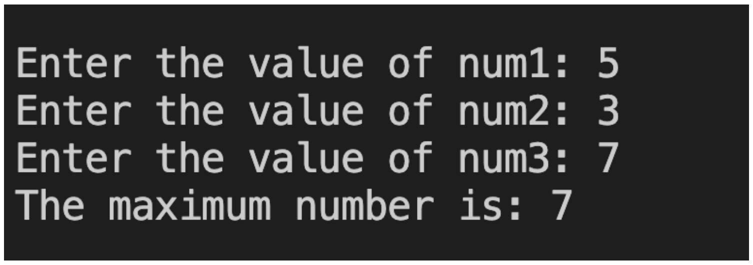
```
Welcome to My Calculator!

Enter the first number: 6
Enter the second number: 2
Select an operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division

Enter the number corresponding to your choice: 3
Result: 12
```

Task 05

Write a C++ Program to find maximum of three numbers using functions. Write a function which takes 3 integers as arguments and return the max number out of the 3. The prototype of the function should be:

int computeMax(int num1, int num2, int num3)

Output:



```
Enter the value of num1: 5
Enter the value of num2: 3
Enter the value of num3: 7
The maximum number is: 7
```