

CL1002-Programming Fundamentals— FALL 2023

LAB 9

Loops + Functions



Learning Outcomes

In this lab you are expected to learn the following:

- Nested Loops (Revision)
- Functions



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);
```

```
    ... ..
```

```
}
```

function
call

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_LAB09_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

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 02

Write a C++ program that allows the user to input a series of integers until a negative number is entered. Implement a function **bool CheckEvenOdd(int num)** to determine whether a given number is odd or even. Display the count of odd and even numbers at the end.

Output:



```
Enter a list of numbers, terminated by -1:
Enter number : 5
Enter number : 12
Enter number : 7
Enter number : 8
Enter number : -1
Number of even numbers : 2
Number of odd numbers : 2
```

Task 03

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
```

Task 04

Write a program to compute $\sin x$ for given x using functions. The user should supply x and a positive integer n to the function and the function should return a result in double. We compute the \sin of x using the series and the computation should use all terms in the series up through the term involving x^n

$$\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + \dots x^n/n!$$

The series requires the factorial values. Write a separate function for this purpose which returns the factorial of the given number. Use this function within the function used for calculating $\sin x$. The prototype of functions should be:

int factorial(int num)

double sin(int x, int n)



Output:

```
Enter the value of x: 1
Enter the value of n: 10
sin(1) = 0.841471
```

Task 05

Write a program that evaluates the following formula using the concept of function overloading. The program should have functions that take q1 and q2 as double as well as integer. Both the functions should return value in double. Pi and Epsilon should be passed as default arguments in both functions.

double evaluate(int x, int y, double pi=3.142, double epsilon=8.85418)

double evaluate(double x, double y, double pi=3.142, double epsilon=8.85418)

The first function should return value accurate to 3 decimal places and second function should return value accurate to 4 decimal places.

$$F = \frac{|q_1 q_2|}{4\pi\epsilon_0 r^2}$$

Output:

```
Enter the value of q1: 2
Enter the value of q2: 3
Enter the value of r: 2
F = 0.0134796
```

```
Enter the value of q1: -2
Enter the value of q2: 3
Enter the value of r: 2
F = 0.0134796
```