National University
of computer and emerging sciences

LAB 06

Summary

Items	Description
Course Title	Programming Fundamentals
Lab Title	Operators in C++
Duration	3 Hours
Operating System/Tool/Language	Ubuntu/ g++/ C++
Objective	To get familiar with use of Functions in C++ (Definition, Calling, Forward Declaration)

- **Functions**

A function is a collection of statements 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. These small functions can then be executed in the desired order to solve the problem. This approach is sometimes called divide and conquer because a large problem is divided into several smaller problems that are easily solved. The figure 1. illustrates this idea by comparing two programs: one that uses a long complex function containing all of the statements necessary to solve a problem, and another that divides a problem into smaller problems, each of which are handled by a separate function.

This program has one long, complex function containing all of the statements necessary to solve a problem.

```
int main()
{
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
   statement;
}
```

In this program the problem has been divided into smaller problems, each of which is handled by a separate function.

```
int main()
{
   statement;
   statement;
   statement;
}
```

main function

```
void function2()
{
   statement;
   statement;
   statement;
}
```

function 2

```
void function3()
{
   statement;
   statement;
   statement;
}
```

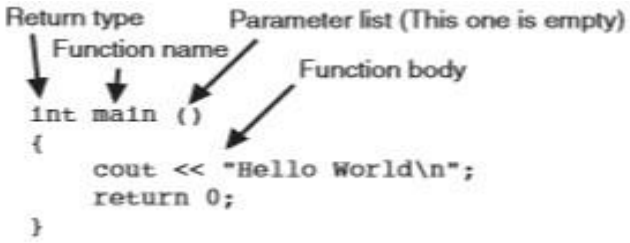
function 3

```
void function4()
{
   statement;
   statement;
   statement;
}
```

function 4

Defining a function

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:



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.

Name: You should give each function a descriptive name. In general, the same rules that apply to variable names also apply to function names.

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.

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.

Void Functions

It isn't necessary for all functions to return a value. These are called void functions. The displayMessage function, which follows, is an example.

```
void  displayMessage()
{ cout << "Hello from the function display Message.\n";
            }
```

The return Statement

The return statement causes a function to end immediately. When the last statement in a void 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. This is demonstrated in Program 1.

Program 1

```
1  // This program uses a function that returns a value.
2  #include <iostream>
3  using namespace std;
4
5  // Function prototype
6  int sum(int, int);
7
8  int main()
9  {
10     int value1 = 20,     // The first value
11         value2 = 40,     // The second value
12         total;           // To hold the total
13
14     // Call the sum function, passing the contents of
15     // value1 and value2 as arguments. Assign the return
16     // value to the total variable.
17     total = sum(value1, value2);
18
19     // Display the sum of the values.
20     cout << "The sum of " << value1 << " and "
21          << value2 << " is " << total << endl;
22     return 0;
23 }
24
25 //*********************************************************
26 // Definition of function sum. This function returns *
27 // the sum of its two parameters.                     *
28 //*********************************************************
29
30 int sum(int num1, int num2)
31 {
32     return num1 + num2;
33 }
```

Program Output

```
The sum of 20 and 40 is 60
```

- Calling Function

A function is executed when it is called. Function main is called automatically when a program starts, but all other functions must be executed by function call statements. When a function is called, the program branches to that function and executes the statements in its body. Let's look at below Program 2, which contains two functions:

main and displayMessage .

Program 2

```
1  // This program has two functions: main and displayMessage
2  #include <iostream>
3  using namespace std;
4
5  //*****************************************
6  // Definition of function displayMessage  *
7  // This function displays a greeting.     *
8  //*****************************************
9
10 void displayMessage()
11 {
12     cout << "Hello from the function displayMessage.\n";
13 }
14
15 //*****************************************
16 // Function main                          *
17 //*****************************************
18
19 int main()
20 {
21     cout << "Hello from main.\n";
22     displayMessage();
23     cout << "Back in function main again.\n";
24     return 0;
25 }
```

Program Output

```
Hello from main.
Hello from the function displayMessage.
Back in function main again.
```

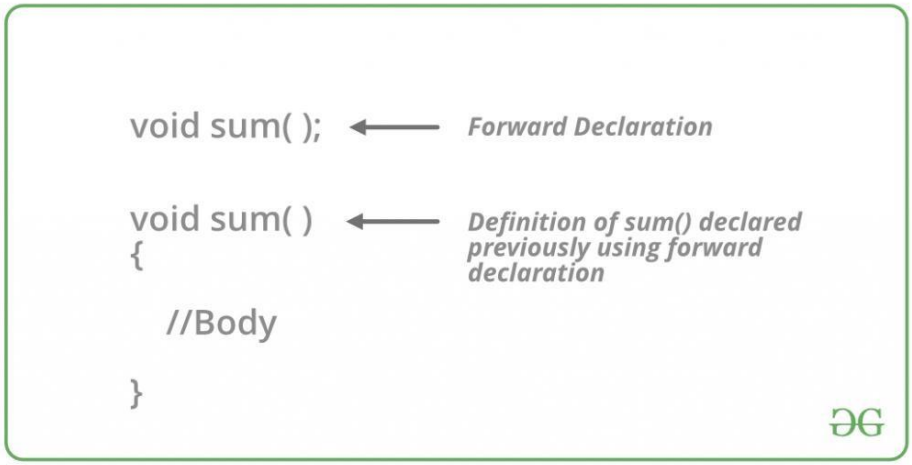
The function displayMessage is called by the following statement in line 22:
displayMessage(); This statement is the function call.

- **Function Header** void displayMessage()
  The function header is part of the function definition. It declares the function's return type, name, and parameter list. It is not terminated with a semicolon because the definition of the function's body follows it.

- **Function Call** displayMessage();
  The function call is a statement that executes the function, so it is terminated with a semi- colon like all other C++ statements. The return type is not listed in the function call, and, if the program is not passing data into the function, the parentheses are left empty.

 Forward Declaration of 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.



Program 3

```
#include <iostream>
Using namespace std;
int add(int x, int y);
// forward declaration of add() (using a function prototype)
int main()
{  cout << "The sum of 3 and 4 is: " << add(3, 4) << '\n';
// this works because we forward declared add() above
return 0;
}
int add(int x, int y)
// even though the body of add() isn't defined until here
{     return x + y;
}
```

Now when the compiler reaches the call to add in main, it will know what add looks like (a function that takes two integer parameters and returns an integer), and it won't complain. It is worth noting that function prototypes do not need to specify the names of the parameters. In the above code, you can also forward declare your function like this:

```
int add(int, int); // valid function prototype
```

Function prototypes are usually placed near the top of a program so the compiler will encounter them before any function calls.

National University
of computer and emerging sciences

Program 4

```
// This program has three functions: main, first, and second.
#include <iostream>
using namespace std;

//*****************************************
// Definition of function first           *
// This function displays a message.      *
//*****************************************

void first()
{
    cout << "I am now inside the function first.\n";
}

//*****************************************
// Definition of function second          *
// This function displays a message.      *
//*****************************************

void second()
{
    cout << "I am now inside the function second.\n";
}

//*****************************************
// Function main                          *
//*****************************************

int main()
{
    cout << "I am starting in function main.\n";
    first();    // Call function first
    second();   // Call function second
    cout << "Back in function main again. \n";
    return 0;
}
```

Program Output

```
I am starting in function main.
I am now inside the function first.
I am now inside the function second.
Back in function main again.
```

Lab Tasks

Task#01

Write a C++ program to print the following:

It was narrated from Umm Salamah that when the Prophet (ﷺ) performed the Subh (morning prayer), while He (PBUH) said the Salam, He (PBUH) would say:

'Allahumma inni as'aluka' ilman nafi'an, wa rizqan tayyiban, wa 'amalan mutaqabbalan (O Allah, I ask You for beneficial knowledge, good provision and acceptable deeds).'

— Sunan of Ibn Majah, Vol. 1, Book 5, Hadith 925

National University
of computer and emerging sciences

Task#02

Write a program in C++ that contains a function to print the upper and lower limits of integer.

- The function should be declared forward.
- The function contains statements to print upper and lower limit of integer data type.
- The function should be called from main.
- **The output should be in following format:**

```
The upper limit of integer is: 2147483647
The lower limit of integer is: -2147483648
```

Task#03

Write a program in C++ that contain a function to calculate volume of a sphere:

- The program should take radius from user.
  - The function should take radius as an input parameter
- The function should be called from main.
  - The function should return value of volume to main.
  - The program prints volume in main.

Task#04

Write a program in C++ that contains a function named timesTen . When timesTen is called, it should display the product of number times ten.

- The function should be declared forward.
- The program should take number from user.
- The function takes value of number as an input parameter.
- The function should calculate ten times of that integer.
- The function should display product of number times ten.

Task#05

Write a program in C++ that contains a function to calculate third angle of triangle.

- The program should take two angles from user.
- The function should take two angles as an input parameter.
- The function will calculate the third angle and return it to main.
- The program should print value of third angle in main.

Practice Question:
Run all sample programs

Submission Instructions:

1. Save all .cpp files with your roll no and task number
   e.g. i21XXXX_Task01.cpp
2. Save all screenshots of terminal with your roll no and task number
3. Now create a new folder with name ROLLNO_LAB03 e.g. i21XXXX_LAB03
4. Move all your .cpp files to this newly created directory and compress it into .zip file.
5. Now you must submit this zipped file on Google Classroom.

OR
You can make a single file where you will be pasting all your solutions with screenshots.