RIPHAH INTERNATIONAL UNIVERSITY

Faculty of Computing

Object Oriented Programming Lab

Lab 1

Lab Manual Development Team

Supervision and Coordination

Shazwa Tun Naeem

Lecturer

Faculty of Computing

Lab Designers

Hafsah Mahmood

Teaching Fellow

Faculty of Computing

Lab 1: Functions and Basic concept revision

1. Introduction

Object-Oriented Programming (OOP) is a programming language that organizes code using objects, which are instances of classes. You will notice several objects (things) like pens, tablets, laptops, etc. Things have **attributes** called **characteristics** like color height, color, weight, age, etc.

2. Concept Map

1. Functions (pass by value ,pass by reference)
2. Lab Exercise & Demo.

3. Activity Time boxing

Table 1: Activity Time Boxing

Task No.	Activity Name	Activity time	Total Time
1	Task 1	20 mins	
2.	Task 2.	20 mins	
3.	Task 3	20 mins	
4.	Task 4	20 mins	
5.	Task 5	20 mins	
5.	Lab Exercise & Demo	100 mins	150minutes

4. Functions

4.1. Functions

When you pass arguments to a function, they can be passed using two mechanisms: "pass by value" and "pass by reference."

1. Functions without parameters and return type

```
void greet() {  // Function definition
   cout << "Hello! Welcome to C++ Functions." << endl;
}
```

2. Functions with Parameters

```
void add(int a, int b) {  // Function definition with parameters
   cout << "Sum: " << a + b << endl;
}
```

3. Functions with Return Types

```
int multiply(int x, int y) {  // Function returning a value
   return x * y;
}
```

Noted: Functions cannot be defined inside other functions.

4.2. Structured Arrays

This program:

1. Uses a **structure array** to store multiple student records.
2. Implements the following **functions**:
   - `InputStudentData()`: To take student details as input.
   - `DisplayStudentData()`: To display all stored student records.
   - `FindTopStudent()`: To find and display the student with the highest marks.

```
#include <iostream>
#include <string>


using namespace std;


// Structure to store student details
struct Student {
    string name;
    int rollNumber;
    float marks;
```

```
};

// Function to input student data
void InputStudentData(Student students[], int count) {
    for (int i = 0; i < count; i++) {
        cout << "\nEnter details for Student " << i + 1 << ":\n";
        cout << "Name: ";
        cin >> students[i].name;
        cout << "Roll Number: ";
        cin >> students[i].rollNumber;
        cout << "Marks: ";
        cin >> students[i].marks;
    }
}

// Function to display student records
void DisplayStudentData(Student students[], int count) {
    cout << "\n--- Student Records ---\n";
    for (int i = 0; i < count; i++) {
        cout << "Roll Number: " << students[i].rollNumber << "\n";
        cout << "Name: " << students[i].name << "\n";
        cout << "Marks: " << students[i].marks << "\n\n";
    }
}

// Function to find the student with the highest marks
void FindTopStudent(Student students[], int count) {
    int topIndex = 0;
    for (int i = 1; i < count; i++) {
        if (students[i].marks > students[topIndex].marks) {
            topIndex = i;
        }
    }
    cout << "\n--- Top Student ---\n";
    cout << "Name: " << students[topIndex].name << "\n";
    cout << "Roll Number: " << students[topIndex].rollNumber << "\n";
    cout << "Marks: " << students[topIndex].marks << "\n";
}

int main() {
    int numStudents;
    cout << "Enter the number of students: ";
    cin >> numStudents;

    Student students[numStudents]; // Structured array

    InputStudentData(students, numStudents);
    DisplayStudentData(students, numStudents);
    FindTopStudent(students, numStudents);

    return 0;
}
```

Lab task 1

You are required to develop a **C++ program** that performs various measurement conversions using **functions**. The program should:

Display a menu with the following options:

- Convert **Kilometers to Miles** *(1 Km = 0.621 Miles)*
- Convert **Celsius to Fahrenheit** *(F = (C × 9/5) + 32)*
- Convert **Seconds into Hours, Minutes & Seconds**

Take user input for the desired conversion type.

Call the corresponding function to perform the conversion and display the result.

Lab task 2

You are required to develop a **C++ program** that calculates an employee's total salary based on basic pay, allowances, and deductions using **functions**.

Ask the user to enter:

- **Basic salary**
- **Allowance percentage** (e.g., 20% of basic salary)
- **Deduction percentage** (e.g., 10% of basic salary)

Use **functions** to:

- Calculate the **gross salary** (Basic + Allowance)
- Calculate the **net salary** (Gross - Deduction)

Display the **final net salary**.

Lab task 3

You are required to develop a **C++ program** that manages **employee records** using **structure arrays and functions**. The program should:

Ask the user for the **number of employees**.

Use a **structure array** to store multiple employee records.

Implement the following **functions**:

- **InputEmployeeData()** → To take employee details as input.
- **DisplayEmployeeData()** → To display all stored employee records.
- **FindHighestSalary()** → To find and display the employee with the highest salary.

5. Further Reading

5.1. Books

The slides and reading material can be accessed from the folder of the class instructor available at Moellium.