University of Central Punjab

**Faculty of Information Technology**

**Object Oriented Programming**

**Fall 2025**

**Lab 02**

**Topic**

Functions

Pass-by-Value, Pass-by-Reference

Function Overloading, Default Arguments

\* Dynamic Memory Allocation, Dangling Pointers, and Memory Leakage

**Objective**

The basic purpose of this lab is to revise some preliminary concepts of C++ that has been covered in the course of Introduction to Computing and Programming Fundamentals.

Its objective is to recall previously learned basic concepts like revision of arrays and functions

**CLO**

Apply object-oriented programming principles to implement real-world problems.

**Instructions:**

• Indent your code.

• Comment your code.

• Use meaningful variable names.

• Plan your code carefully on a piece of paper before you implement it.

• Name of the program should be same as the task name. i.e. the first program should be Task\_1.cpp

**• void main() is not allowed. Use int main()**

**• You are not allowed to use any built-in functions**

**• You are required to follow the naming conventions as follow:**

**• Variables:** firstName; (no underscores allowed)

**• Function:** getName(); (no underscores allowed)

**• ClassName:** BankAccount (no underscores allowed)

**Students are required to complete the following tasks in lab timings.**

**Task 1:**

Write a C++ function to swap two integers. Implement this function in two ways:

• One using pass-by-value

• One using pass-by-reference

**void swapByValue(int a, int b);**

**void swapByReference(int &a, int &b);**

In main function, demonstrate each version of the **‘Swap** function and print the results.

**Task 2:**

Write an overloaded function named **‘average’** that calculates:

• Average of two **integers**.

• Average of two **doubles**.

You can use return type for functions of your own choice (void, int, double).

**void average(int a, int b);**

**void average(double a, double b);**

In the main function, demonstrate each version of the **‘average’** function and print the results.

**Task 3:**

Write an overloaded function named **‘Area’** that will calculate:

• Area of a **Square**.

• Area of a **Triangle**.

• Area of a **Circle**.

You can use return type for functions, of your own choice (void, double, int).

**void Area(int length);**

**void Area(int length, int width);**

**void Area(double radius);**

In the main function, demonstrate each version of the **‘Area’** function and print the results.

**Task 4:**

Write a C++ program that defines a function calculate that computes the area of a rectangle.

The function should accept two arguments:

• length

• width

The width argument should have a default value of 5.

**int area1 = calculate(10); // Uses default width = 5**

**int area2 = calculate(10, 3); // Uses provided width = 3**

Demonstrate how the function behaves when the width is provided and when it defaults to 5.

**Task5:**

Write a C++ program that:

• Dynamically allocates memory for an array of 5 integers.

• Assigns values to the array.

• Demonstrates the problem of dangling pointers by deleting the memory and trying to access the deleted pointer.

• Shows an example of memory leakage where memory is allocated but not properly deallocated before the program exits.

**Task 6:**

Now create a menu based program where user should select the function that needs to be tested from the above functions.

**For example:**

• Press 1 for PassByValue and PassByRef Function

• Press 2 for Overloaded Average Function

• Press 3 for Overloaded Area Function

• Press 4 for Default Argument **Calculate** Function.

• Press 5 for Testing Dynamic memory allocation, dangling pointers, and memory leak.

• Press 0 for exit.