

Lab 5

Items	Description
Course Title	Object Oriented Programming
Lab Title	Intro to CLasses
Duration	3 Hours
Tools	Eclipse/ C++
Objective	To get familiar with the use of CLasses in c++

Introduction to Classes

A **class** in C++ is a blueprint for creating objects. It defines the properties (data members) and functions (member functions or methods) that an object of that class will have. A **class** encapsulates data and methods, enforcing data hiding and abstraction principles of Object-Oriented Programming (OOP).

```
class ClassName {  
    // Data members  
    // Member functions (methods)  
};
```

```
class Student {  
    string rollNo;  
    string name;  
    float gpa;  
};
```

What is an Object?

An **object** is an instance of a class. Once a class is defined, we can create multiple objects that share the class's blueprint. Each object will have its own copy of data members defined in the class but can call the same methods.

ClassName obj; // Creating an object of ClassName

```
Student student1;
```

Accessing Members:

Use the dot operator (.) to access members of a structure.

```
cout << "Student Name: " << student1.name << endl;
```

Constructors

A **constructor** is a special member function that is automatically called when an object is created. It typically initializes the object's data members. Constructors do not have a return type and share the same name as the class.

```
class MyClass {  
public:  
    int x;  
  
    // Constructor  
    MyClass() {  
        x = 0; // Default initialization  
    }  
};
```

Types of Constructors:

1. **Default Constructor:** A constructor with no parameters.
2. **Parameterized Constructor:** A constructor that takes arguments to initialize object data.
3. **Copy Constructor:** A constructor that initializes an object using another object of the same class.

Access Modifiers

Access modifiers control the visibility of class members. There are three access levels:

1. **Public:** Members declared as **public** are accessible from outside the class.
2. **Private:** Members declared as **private** are accessible only within the class.
3. **Protected:** Members declared as **protected** are accessible within the class and by derived classes.

```
class MyClass {  
private:  
    int secret; // Accessible only within the class  
  
public:  
    int publicInfo; // Accessible from outside  
  
protected:  
    int protectedInfo; // Accessible in the class and  
derived classes  
};
```

```
#include <iostream>
```



```
using namespace std;

class MyClass {
private:
    int privateVar;

public:
    // Default constructor
    MyClass() {
        privateVar = 10;
    }

    // Parameterized constructor
    MyClass(int x) {
        privateVar = x;
    }

    // Public method to access private member
    int getPrivateVar() {
        return privateVar;
    }
};

int main() {
    MyClass obj1; // Calls default constructor
    MyClass obj2(50); // Calls parameterized constructor

    cout << "Object 1 privateVar: " <<
obj1.getPrivateVar() << endl;
```

```
cout << "Object 2 privateVar: " <<  
obj2.getPrivateVar() << endl;  
  
return 0;  
}
```

Lab Tasks

Task 1

Write a complete C++ program with the following features.

- A. Declare a class Point with two integer data members x and y.
- B. Provide appropriate constructors to initialize the data members.
- C. Provide a method display() to print the x and y coordinates.
- D. In the main(), declare two objects of Point and call the method display() for each of them.

Task 2

Define a class Circle with its radius and center(x,y) as data members. Provide appropriate constructors and methods to compute the area and circumference of the circle. Call these methods from the main() program.

Task 3

Build a class Sale with private member variables double itemCost;

// Cost of the item double taxRate;

// Sales tax rate and functionality mentioned below:

- Write a default constructor to set the member variable itemCost to 0 and taxRate to 0.
Sale()
- Write a parameterized constructor that accepts the parameter for each member variable such as cost for ItemCost and rate for taxRate Sale(double cost, double rate)
- Write a function getTax() to calculate tax i.e take a product of itemCost and itemRate.
double getTax()
- Write a function getTotal() to calculate the total price of item i.e. take a sum of itemCost and getTax() (calling getTax() will return the calculated tax on item) . double getTotal()