### **Lab3: Class & Object, Constructor and Destructor in C++**

### **Objective:**

* To understand the basic concepts of Class and Object in C++.
* To learn how constructors initialize objects automatically.
* To understand the purpose and use of destructors for cleanup.
* To implement a simple C++ program using classes, constructors, and destructors.

### **Theory:**

#### **1. Class and Object**

* A class is a user-defined data type that acts as a blueprint for creating objects.
* An object is an instance of a class that can access the class’s public members.

**Syntax:**

class ClassName {

public:

int data; // Data Member

void display() { // Member Function

cout << "Data: " << data << endl;

}

};

int main() {

ClassName obj; // Creating object

obj.data = 10;

obj.display();

return 0;

}

#### **2. Constructor**

* A **constructor** is a special function that is called automatically when an object is created.
* It has the same name as the class and no return type.
* It is used to initialize objects.

**Types of constructors:**

* **Default Constructor** – No parameters.
* **Parameterized Constructor** – Takes arguments.
* **Copy Constructor** – Initializes an object using another object.

**Syntax:**

class Example {

public:

Example() { // Default Constructor

cout << "Default Constructor Called" << endl;

}

Example(int a) { // Parameterized Constructor

cout << "Parameterized Constructor Called: " << a << endl;

}

};

#### **3. Destructor**

* A **destructor** is a special function that is automatically invoked when an object is destroyed.
* It is used to free resources and perform clean-up operations.
* Its name is the same as the class but prefixed with a tilde ~.
* It takes no arguments and has no return type.

**Syntax:**

class Example {

public:

~Example() {

cout << "Destructor Called" << endl;

}

};