OOP’S ASSIGNMENT – I (C++)

**NAME : S.Harshini**

**DEPT : AI & DS**

**YEAR : II – 3rd SEM**

**SUBMITTED ON : 21.07.2025**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PROGRAM 1: RECTANGLE CLASS**

Develop a C++ code to create a class Rectangle and object and print the member variables along with the area.

**CODE :**  
#include <iostream>  
using namespace std;  
  
class Rectangle {  
public:  
 int length, width;  
  
 void setData(int l, int w) {  
 length = l;  
 width = w;  
 }  
  
 void display() {  
 cout << "Length: " << length << endl;  
 cout << "Width: " << width << endl;  
 cout << "Area: " << length \* width << endl;  
 }  
};  
  
int main() {  
 Rectangle r1;  
 r1.setData(5, 10);  
 r1.display();  
 return 0;

}

**Output:**  
Length: 5  
Width: 10  
Area: 50

# PROGRAM 2: CONSTRUCTORS DEMONSTRATION

Develop a C++ program to show the working of default constructor, parameterized constructor, and copy constructor, and destruct any object.

**CODE :**  
#include <iostream>  
using namespace std;  
  
class MyClass {  
 int value;  
  
public:  
   
 MyClass() {  
 value = 0;  
 cout << "Default constructor called. Value = " << value << endl;  
 }  
  
   
 MyClass(int val) {  
 value = val;  
 cout << "Parameterized constructor called. Value = " << value << endl;  
 }  
  
   
 MyClass(const MyClass &obj) {  
 value = obj.value;  
 cout << "Copy constructor called. Value = " << value << endl;  
 }  
  
   
 ~MyClass() {  
 cout << "Destructor called for value = " << value << endl;  
 }  
};  
  
int main() {  
 MyClass obj1;   
 MyClass obj2(100);   
 MyClass obj3 = obj2;   
 return 0;  
}

**Output:**  
Default constructor called. Value = 0  
Parameterized constructor called. Value = 100  
Copy constructor called. Value = 100  
Destructor called for value = 100  
Destructor called for value = 100  
Destructor called for value = 0

# PROGRAM 3: STATIC MEMBER COUNTER

Create a C++ program with a class Counter that has a static member Count. Track the number of objects created. Implement a static getCount() to return this count. In the main function, create multiple Counter objects and display the count.

**CODE :**  
#include <iostream>  
using namespace std;  
  
class Counter {  
 static int count;  
  
public:  
 Counter() {  
 count++;  
 }  
  
 static int getCount() {  
 return count;  
 }  
};  
  
int Counter::count = 0;  
  
int main() {  
 Counter c1, c2, c3;  
 cout << "Total objects created: " << Counter::getCount() << endl;  
 return 0;  
}

**Output:**  
Total objects created: 3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_