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

Program :

```
#include <iostream>

using namespace std;

class Rectangle {

public:

   int length;

   int breadth;


   int area() {

      return length * breadth;

   }


   void display() {

      cout << "Length: " << length << endl;

      cout << "Breadth: " << breadth << endl;

      cout << "Area: " << area() << endl;

   }

};


int main() {

   Rectangle obj;

   obj.length = 10;

   obj.breadth = 7;

   obj.display();

   return 0;

}
```

Output :

Length: 10

Breadth: 7

Area: 70

=== Code Execution Successful ===

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

Program :

```
#include <iostream>
class Student {
   int rollNo;
   std::string name;
public:
   // Default constructor
   Student() {
      rollNo = 0;
      name = "Unknown";
      std::cout << "Default constructor \n";
   }
   // Parameterized constructor
   Student(int r, std::string n) {
      rollNo = r;
      name = n;
      std::cout << "Parameterized constructor \n";
   }
   // Copy constructor
   Student(const Student &s) {
      rollNo = s.rollNo;
      name = s.name;
      std::cout << "Copy constructor \n";
   }
```

```
    // Display function
    void display() {
        std::cout << "Roll No: " << rollNo << ", Name: " << name << "\n";
    }
    // Destructor
    ~Student() {
        std::cout << "Destructor called for " << name << "\n";
    }
};
int main() {
    Student s1;                // Default constructor
    s1.display();
    Student s2(42, "santhosh");   // Parameterized constructor
    s2.display();
    Student s3 = s2;           // Copy constructor
    s3.display();

    return 0;
}
```

Output :

```
Default constructor
Roll No: 0, Name: Unknown
Parameterized constructor
Roll No: 42, Name: santhosh
Copy constructor
Roll No: 42, Name: santhosh
Destructor called for santhosh
Destructor called for santhosh
Destructor called for Unknown


=== Code Execution Successful ===
```

3.Create a C++ program with a class Counter that has a static member count to track the number of objects created.

Program :

```
#include <iostream>

class Counter {
   static int count;
public:
   Counter() {
      count++;
      std::cout << "Object . Count: " << count << std::endl;
   }
   static int getCount() {
      return count;
   }
};

int Counter::count = 0;

int main() {
   std::cout << "Starting count: " << Counter::getCount() << std::endl;

   Counter a, b, c;

   std::cout << "Total objects: " << Counter::getCount() << std::endl;

   return 0;
```

}

Output :

Starting count: 0

Object . Count: 1

Object . Count: 2

Object . Count: 3

Total objects: 3

=== Code Execution Successful ===