1. C++ Code to Create a class Rectangle and object print the member variables along with the area.

Algorithm:

Step1: Create a class Rectangle with two member variables length, width

Step2: Create a member function display () to: print length, width and area

Step3: Create an object of rectangle in main function

Program:

```
# include <iostream.h>
using namespace std;

class rectangle {
private:
   int length;
   int width;
public:
   rectangle (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(10,5);
    r1. display();
    return 0;
}.
```

Output:

```
length: 10
width: 5
Area: 50
```

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

Algorithm:

Step 1: Create a class person with three data members of age, weight and height

step 2: define it with defaut constructor, paramaterized Constructor and copy constructor from another object

Step 3: Create a destructor to destruct the object

Step 4: Create a display() function to print age, weight and height. and Create main function and define object in it.

Program:

```
# include <iostream.h>
using namespace std;
class person {
private:
    int age;
    float height;
    float weight;
public:
  person() {
   age = 0;
   height = 0.0;
   weight = 0.0;
   cout << "Default constructor called \n";
  }
 person(int a, float h, float w) {
    age = a;
    height = h;
    weight = w;
    cout << "parameterized constructor called \n";

 }
person (const person &p) {
    age = p.age;
    height = p.height;
    weight = p weight;
    cout << "Copy Constructor Called \n";
}
~person () {
    cout << "Destructor called \n";
}
void display () {
    cout << "Age: " << age << "years \n";
    cout << "Height: " << height << "cm\n";
    cout << "Weight: " << weight << "kg\n";
}
};
int main () {
    person p1;
    Person p2 (25, 170.5, 70.0);
    person p3 = p2
    cout << "\n Person1 (Default): \n";
    p1.display();
    cout << "\n Person2 (parameterized): \n";
    p2.display();
    cout << "\n person3 (Copy): \n";
    p3.display();
    return 0;
}.
```

Output:

```
Default Constructor Called
parameterized Constructor Called
Copy Constructor Called
person1 (Default):
 Age: 0 years
 Height: 0.0 cm
 Weight: 0.0 kg.
person2 (parameterized):
Age: 25 years
Height: 170.5 cm
Weight: 70kg
person3 (Copy):
 Age: 25 years
 Height: 170.5 cm
Weight: 70 kg
Destructor Called
Destructor Called
Destructor Called.
```

3. C++ program with a class counter that has a static member count track the number of objects created. Implement a static function get count() to return this count. In the main function, create multiple counter objects and display the count.

Algorithm:

Step 1: Create a class counter with static variable count.

Step 2: Create get count() that returns count.

Step 3: In main() function create multiple counter objects and call and display getcount()

program:

```
#include <iostream.h>
using namespace std;
class counter {
private:
    static int count;
public:
    counter () {
    count ++;
    }
   static int getcount() {
        return count;
   }
};
int counter:: count =0;
int main () {
    counter c1, c2, c3;
    cout << "Number of objects created:" << counter:: getcount () << endl;
    return 0;
}.
```

Output:

```
Number of objects created: 3.
```