

OOPS

Static Class Members

We can define class members static using ***static*** keyword.

Characteristics

When we declare a member of a class as static it means no matter how many objects of the class are created, there is only ***one copy*** of the static member.

A static member is shared by all objects of the class.

All static data is initialized to zero when the first object is created, if no other initialization is present.

We can't put it in the class definition but it can be initialized outside the class as done in the following example by redeclaring the static variable, using the scope resolution operator :: to identify which class it belongs to.

```
#include <iostream>
using namespace std;
class item {
    static int count;
    int number;

public:
    void getdata(int a) {
        number = a;
        count++;
    }
    void getcount() { cout << "Count is : " << count << endl; }
};
int item::count = 7;
int main() {
    item a, b, c;
    a.getcount();
    b.getcount();
    c.getcount();
    a.getdata(100);
    b.getdata(200);
    c.getdata(300);
    cout << "After Reading Data \n";
    a.getcount();
    b.getcount();
    c.getcount();
    return 0;
}
```

Output:

```
Count is : 7
Count is : 7
Count is : 7
After Reading Data
Count is : 10
Count is : 10
Count is : 10
```

Static Function Members

By declaring a function member as static, you make it independent of any particular object of the class.

A static member function can be called even if no objects of the class exist and the static functions are **accessed using only the class name and the scope resolution operator ::**

A static member function can only access static data member, other static member functions and any other functions from outside the class.

Static member functions have a class scope and they do not have access to the this pointer of the class.

```
#include <iostream>
using namespace std;
class Box {
    private:
        double length;
        double breadth;
        double height;

    public:
        static int objectCount;
        Box(double l, double b, double h) {
            cout << "Constructor called." << endl;
            length = l;
            breadth = b;
            height = h;
            objectCount++;
        }
        static int getCount() { return objectCount; }
};
// Initialize static member of class
int Box::objectCount = 0;
int main(void) {
    // Print total number of objects before creating object.
    cout << "Initial Count: " << Box::getCount() << endl;
    Box Box1(1, 2, 3);
    Box Box2(4, 5, 6);
    // Print total number of objects after creating object.
    cout << "Final Count: " << Box::getCount() << endl;
    return 0;
}
```

Output:

```
Initial Count: 0
Constructor called.
Constructor called.
Final Count: 2
```



NO NOTES TO SHOW