Object-Oriented Programming Using C++

Static Class Members

Indranil Saha

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

Static Member

- We can define class members static using static keyword
- 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 by redeclaring the static variable, using the scope resolution operator :: to identify which class it belongs to

Example: Static Member

Destructor for the Line class

```
class Box {
   private:
      double length;      // Length of a box
      double breadth;     // Breadth of a box
      double height;      // Height of a box

   public:
      static int objectCount;

      // Constructor definition
      Box(double l = 2.0, double b = 2.0, double h = 2.0) {
         cout <<"Constructor called." << endl;
         length = l; breadth = b; height = h;
         // Increase every time object is created
         objectCount++;
      }
      double Volume() {
         return length * breadth * height;
      }
};

// Initialize static member of class Box
int Box::objectCount = 0;

int main(void) {
   Box Box1(3.3, 1.2, 1.5);    // Declare box1
   Box Box2(8.5, 6.0, 2.0);    // Declare box2
   // Print total number of objects.
   cout << "Total objects: " << Box::objectCount << endl;
   return 0;
}
```

Example: Static Member

Output

```
Constructor called.
Constructor called.
Total objects: 2
```

Static Member Function

- By declaring a function member as static, you make it independent of any particular object of the class
- The static functions are accessed using only the class name and the scope resolution operator ::.
- Can be called even if no objects of the class exist
- Can only access static data member, other static member functions and any other functions from outside the class
- Does not have access to the this pointer of the class
- You could use a static member function to determine whether some objects of the class have been created or not

Example: Static Member Function

Static Member Function for the Box class

```
class Box {
   private:
      double length;      // Length of a box
      double breadth;     // Breadth of a box
      double height;      // Height of a box
   public:
      static int objectCount;
      // Constructor definition
      Box(double l = 2.0, double b = 2.0, double h = 2.0) {
         cout <<"Constructor called." << endl;
         length = l; breadth = b; height = h;
         objectCount++;
      }
      double Volume() {
         return length * breadth * height;
      }
      static int getCount() {
         return objectCount;
      }
};

int Box::objectCount = 0;

int main(void) {
   cout << "Inital Stage Count: " << Box::getCount() << endl;
   Box Box1(3.3, 1.2, 1.5);    // Declare box1
   Box Box2(8.5, 6.0, 2.0);    // Declare box2
   // Print total number of objects after creating object.
   cout << "Final Stage Count: " << Box::getCount() << endl;
   return 0;
}
```

Example: Static Member Function

Output

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

Object-Oriented Programming Using C++

Static Class Members

Indranil Saha

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur