

Aim: Practice on class definitions, static class members and friend functions.

1. Create a class called **Box** for a cubic box with 3 data members, **color**, **length** and **volume**. Color indicates the color of the box, length indicates the length of an edge of the box, and volume indicates the volume of the box.

2. Define constructors and destructors.

Default constructor: Define a constructor (ctor) without parameters.

Parameterized constructor: Define a constructor with parameters that initialize the data members.

Destructor: Define also a destructor (dtor) for the class.

3. Define **set** and **get** methods for the data members of the class. Make sure that the **get** method is **const**. While implementing the **set** method, use proper **validation** mechanisms on the input data.

4. Define a member function **calculateVolume()** that calculates volume of the box. Call this function in both default and parameterized constructors to calculate the volume.

5. Extend your class in the following way: Add an extra **private static** data member called **numberOfBoxes** to count the number of Box objects in the memory. Make sure that you initialize **numberOfBoxes** as 0, increment it in the constructors and decrement it in the destructor. Define also a **public static get** method to return the value of **numberOfBoxes**.

6. Define a non-member (top-level) function that prints color, length and volume members of a given Box object. Use **friend** keyword to provide access to the private data.

friend void printBox(Box &); // prototype

7. In the **main()** function, create an array of objects for storing 5 boxes. Ask user to enter the **color** and **length** for these 5 boxes. After calculating the volumes of these boxes, use the **printBox** function to print their information. Before and after printing them, also print the number of boxes. You can also create instances using **new** operator (dynamically at run time). Then you can print the number of boxes again. After creating some instances, remove some of them from the memory by using **delete** operator and observe the number of boxes after this change.