

Assignment-1

In the following example, the Box class contains three private member variables width, height and depth; four overloaded versions of constructors to initialize the variables, getter and setter methods to read and write these the private variables, a volume method to determine the volume of a box object.

Another class BasicOOP_Demo has been created with the main method to demonstrate the execution of this program including object creation and other operations. Output of this example program is also given at the end for understanding the result. Obviously, the following example design and implementation have been used to demonstrate, and you cannot use this as your solution.

Your task will be to design and implement **any real-life class** similar to the following example with **meaningful** name, member variables and methods. Don't use any class design that has already been taught such as Box, Stack and Student.

Moreover, take care of the followings

- I. Declare all the member variables private to maintain encapsulation. Design appropriate public methods to access/modify them.
- II. Perform method overloading by either using constructor or any other meaningful member method.
- III. Design at least one method other than constructors (such as volume method in the given example) to perform some useful calculations on the objects.
- IV. Design another class (such as BasicOOP_Demo) and the main method inside it. Declare necessary objects of your designed class.
- V. Show method overloading using all the overloaded versions you have designed. And, perform some calculations on the objects using the designed method (such as calculating volume of a box object using the volume method in the given example). Use also some of the other public methods to access/modify the values of any object (e.g., get and set methods are used in the given example).

```
class Box{

    private double width,height,depth;

    Box(){ }
    Box(double w,double h,double d){
        width = w;
        height = h;
        depth = d;
    }
    Box(double val){
        width = height = depth = val;
    }
    Box(Box obj){
        width = obj.width;
        height = obj.height;
        depth = obj.depth;
    }
}
```

```

public double volume() {
    return width*height*depth;
}
public double getWidth() {
    return width;
}
public void setWidth(double width) {
    this.width = width;
}
public double getHeight() {
    return height;
}
public void setHeight(double height) {
    this.height = height;
}
public double getDepth() {
    return depth;
}
public void setDepth(double depth) {
    this.depth = depth;
}
}

public class BasicOOP_Demo {

    public static void main(String[] args) {

        Box myBox1 = new Box(2,3,4);
        Box myBox2 = new Box();
        Box myCube = new Box(5);
        Box myClone = new Box (myBox1);

        System.out.println("Volume of myBox1: "+myBox1.volume());
        System.out.println("Volume of myBox2: "+myBox2.volume());
        System.out.println("Volume of myBoxCube: "+myCube.volume());
        System.out.println("Volume of myClone: "+myClone.volume());

        myBox2.setWidth(10);
        myBox2.setHeight(15);
        myBox2.setDepth(20);
        System.out.println("Modified values of myBox2 attributes: "
            +myBox2.getWidth()+" "+myBox2.getHeight()+" "+myBox2.getDepth());

        System.out.println("Volume of myBox2: "+myBox2.volume());

    }

}

```

OutPut:

```

Volume of myBox1: 24.0
Volume of myBox2: 0.0
Volume of myBoxCube: 125.0
Volume of myClone: 24.0
Modified values of myBox2 attributes: 10.0 15.0 20.0
Volume of myBox2: 3000.0

```