

# **OOP Practical-6**

**Date:** 06-10-2021

**Name:** Mufid Vahora

**Roll No.:** 20BCE307

**Course Code:** 2CS302

**Course Name:** Object Oriented Programming

## **Practical No. 6a.**

**Aim:** Define a class Rectangle with its length and breadth.

Provide appropriate constructor(s), which gives facility of constructing rectangle object with default values of length and breadth as 0 or passing value of length and breadth externally to constructor.

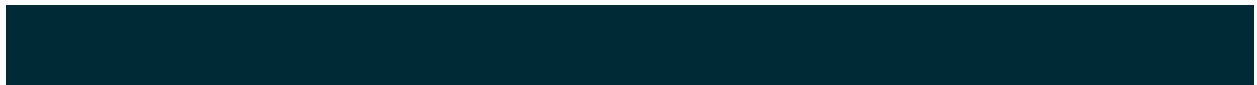
Provide appropriate accessor & mutator methods to Rectangle class.

Provide methods to calculate area & to display all information of Rectangle.

Design different class TestRectangle class in separate source file, which will contain main function. From this main function, create 5 Rectangle objects by taking all necessary information from the user.

The class has attributes length and width, each of which defaults to 1. It should have member functions that calculate the perimeter and area of the rectangle. It should have set and get functions for both length and width. The set functions should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0.

## **Methodology Followed:**



```
//1st file

import java.util.Scanner;

public class Rectangle {

    float l;

    float b;

    public Rectangle() {

        l = 0;

        b = 0;

    }

    public void setter(float x, float y) {

        if ((x >= 0 && x <= 20) && (y >= 0 && y <= 20)) {

            this.l = x;

            this.b = y;

        } else {

            System.out.println("Enter values in range of 0 and 20");

            return;

        }

    }

    public float accl() {

        return l;

    }

}
```

```

public float accb() {

    return b;

}

float area() {

    return l * b;

}

float peri() {

    return 2 * (l + b);

}

public void display() {

    System.out.println("Length=" + l);

    System.out.println("Breadth=" + b);

    System.out.println("Area=" + area());

    System.out.println("Perimeter=" + peri());

}

}

// 2nd file

class TestRectangle {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        Rectangle[] r = new Rectangle[5];

        for (int i = 0; i < 5; ++i) {

```

```
        r[i] = new Rectangle();
    }

    for (int i = 0; i < 5; ++i) {

        float l;

        float b;

        System.out.println("Enter details of object " + (i + 1) + ":");

        l = sc.nextFloat();

        b = sc.nextFloat();

        r[i].setter(l, b);

    }

    System.out.println("-----");
    System.out.println("-----");

    for (int i = 0; i < 5; ++i) {

        System.out.println("The values in object " + (i + 1) + "
are:");

        r[i].display();

    }

    System.out.println("*****");

}

}
```

## Input/Output:

```
Enter details of object1:
5
6
Enter details of object2:
2
3
Enter details of object3:
1
7
Enter details of object4:
9
5
Enter details of object5:
4
6
-----
The values in object 1 are:
Length=5.0
Breadth=6.0
Area=30.0
Perimeter=22.0
*****
The values in object 2 are:
Length=2.0
Breadth=3.0
Area=6.0
Perimeter=10.0
*****
The values in object 3 are:
Length=1.0
Breadth=7.0
Area=7.0
Perimeter=16.0
*****
```

```
The values in object 4 are:
Length=9.0
Breadth=5.0
Area=45.0
Perimeter=28.0
*****
The values in object 5 are:
Length=4.0
Breadth=6.0
Area=24.0
Perimeter=20.0
*****
```

### Practical No. 6b.

**Aim:** Create a class Term. This class represents a term of a polynomial such as  $2x^4$  where 2 is coefficient and 4 is exponent of the term.

Data members:-

- int coefficient
- int exponent

Create another class Polynomial. The internal representation of a polynomial is an array of Terms. The size of this array should be fixed.

Provide a constructor for this class that will set all terms of a polynomial object as zero (where coefficient is 0 and exponent is 0). Provide following functions:

■ `setTerm(int, int)` – Setting a term of a polynomial object. Each successive call of this function should set next term of the polynomial object.

It should do the following validations:-

- Whether the exponent of the term being set is already used.
- Whether the array size limit is exceeded.
- Whether the exponent is negative.

In all the cases it should not set the term and display an appropriate message.

- `sort()` – to arrange the terms in ascending order of exponents.

🏢 provide a function to print a polynomial object

### Methodology Followed:

```
import java.util.Scanner;

class Term {
    int coef;
    int exp;
}

class Polynomial{
    Term[] t=new Term[5];
    static int i;
    Polynomial()
    {
        for(int j=0;j<5;++j)
        {
            t[j]=new Term();
            t[j].coef=0;
            t[j].exp=0;
        }
        i=0;
    }

    void Setterm(int a, int b)    //function sets the values
    {
        int j=i;
        while(j>=0)
        {
            if(t[j].exp==b)
            {
                System.out.println("You have already used this exponent");
                return;
            }
            j=j-1;
        }
        if(j==5)
        {
            System.out.println("Size limit");
            return;
        }
    }
}
```

```

    }
    if(b<0)
    {
        System.out.println("Negative exponent is not allowed");
        return;
    }
    t[i].coef=a;
    t[i].exp=b;
    sort();
    System.out.println("done");
    i++;
}

void sort()
{
    for(int j=0;j<=i;++j)
    {
        for(int k=j;k<=i;++k)
        {
            if(t[j].exp<t[k].exp)
            {
                int temp=t[j].exp;
                t[j].exp=t[k].exp;
                t[k].exp=temp;
            }
        }
    }
}

void display()
{
    System.out.print("Polynomial is:");
    for(int j=0;j<i;++j)
    {
        System.out.print(t[j].coef + "x^(" + t[j].exp + ") ");
        if(j<i-1)
            System.out.print("+ ");
    }
    System.out.println();
}

public static void main(String[] args)
{

```



```

Scanner sc=new Scanner(System.in);
Polynomial p=new Polynomial();
char ch;
do
{
    System.out.print("Enter the coefficient of the polynomial:");
    int c=sc.nextInt();
    System.out.print("Enter the exponent of the polynomial:");
    int e=sc.nextInt();
    p.Setterm(c,e);
    System.out.print("Do you want to continue:");
    ch=sc.next().charAt(0);
}while(ch=='y');
p.display();
}
}

```

### Input/Output:

```

Enter the coefficient of the polynomial:5
Enter the exponent of the polynomial:3
done
Do you want to continue:y
Enter the coefficient of the polynomial:4
Enter the exponent of the polynomial:2
done
Do you want to continue:n
Polynomial is:5x^(3) + 4x^(2)
PS C:\Users\vohra\Desktop\Java> 

```

## Practical No. 6c.

**Aim:** Create a class called complex for performing arithmetic operations with complex numbers. Use floating point variables to represent the private data of the class. Provide a default constructor that initializes the object with some default values. Provide public member functions for each of the following

- Addition of two complex numbers: It returns the result obtained by adding the respective real parts and the imaginary parts of the two complex numbers.
- Subtraction of two complex numbers: It returns the result obtained by subtracting the respective real parts and the imaginary parts of the two complex numbers.
- display( ) – It displays the complex number in a+bi format.

The output should be displayed as follows:-

Sum of  $a_1+b_1 i$  &  $a_2+b_2 i$  is :  $a_3+b_3 i$

## Methodology Followed:

```
import java.util.Scanner;

public class Complex {

    float r;

    float img;

    Complex() {

        r = 0;

        img = 0;

    }

    void setter(float x, float y) {

        r = x;

        img = y;
```

```
}

public static Complex add(Complex n1, Complex n2) {

    Complex sum = new Complex();

    sum.r = n1.r + n2.r;

    sum.img = n1.img + n2.img;

    return sum;

}

public static Complex sub(Complex n1, Complex n2) {

    Complex dif = new Complex();

    dif.r = n1.r - n2.r;

    dif.img = n1.img - n2.img;

    return dif;

}

public void display() {

    if (this.img >= 0)

        System.out.println("The complex number is:" + this.r + "+" +
this.img + "i");

    else

        System.out.println("The complex number is:" + this.r +
this.img + "i");

}

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    Complex c1 = new Complex();
```

```

        Complex c2 = new Complex();

        System.out.print("Enter real part of first complex number:");

        float r = sc.nextFloat();

        System.out.print("Enter imaginary part of first complex number:");

        float img = sc.nextFloat();

        c1.setter(r, img);

        System.out.println();

        System.out.print("Enter real part of second complex number:");

        r = sc.nextFloat();

        System.out.print("Enter imaginary part of second complex
number:");

        img = sc.nextFloat();

        c2.setter(r, img);

        Complex sum = add(c1, c2);

        Complex dif = sub(c1, c2);

        System.out.print("After addition:");

        sum.display();

        System.out.print("After subtraction:");

        dif.display();

    }
}

```

**Input/Output:**

```
Enter real part of first complex number:5
Enter imaginary part of first complex number:2

Enter real part of second complex number:3
Enter imaginary part of second complex number:4
After addition:The complex number is:8.0+6.0i
After subtraction:The complex number is:2.0-2.0i
PS C:\Users\vohra\Desktop\Java> █
```

### **Practical No. 6d.**

**Aim:** Create an object called GSSArray. (It stands for growable self-sorting array)

This object will manage an array of type int. Create a private variable for an array of type int. In the constructor for this object, take in an int value which will determine the starting size of the array. The constructor should also instantiate the array.

Create a public method called insert, which will take in an int and find the location in the array where it belongs and insert it there. If the array is full, then before inserting the value, method insert should call private method increaseSize, which will create a new array which is double the size of the

current array. Then it will copy the values from the original array into the new array and set the private variable to this new array.

The array should keep track of how many of its indexes are filled. Create a private variable called lastindex which will be equal to the last index of the array that has a value.

Create a public method delete, which will take an int and if will remove the 1st instance of that number in the array. If the number doesn't exist, the method should return false, otherwise it should return true. (Don't forget to update variable lastindex in methods delete and insert.)

### Methodology Followed:

```
import java.util.Arrays;
import java.util.Scanner;

public class GSSArray {

    int[] a;

    int lastindex;

    GSSArray(int size) {

        a = new int[size];

        lastindex = -1;

        for (int i = 0; i < size; ++i)

            a[i] = -1;

    }

    public void insert(int x) {

        if (a.length - 1 == lastindex) {

            increaseSize();

        }

    }

}
```

```

    if (lastindex == -1) {

        a[0] = x;

        lastindex = 0;

        return;

    }

    if (a[lastindex] < x) {

        lastindex++;

        a[lastindex] = x;

        return;

    }

    int j;

    for (j = 0; j <= lastindex; ++j) {

        if (a[j] > x) {

            for (int k = lastindex; k >= j; --k) {

                a[k + 1] = a[k];

            }

            a[j] = x;

            lastindex++;

            return;

        }

    }

}

```

```

void increaseSize() {

    int size = a.length * 2;

    int[] ar = new int[size];

    for (int i = 0; i < a.length; ++i)

```

```

        ar[i] = a[i];

        for (int i = a.length; i < ar.length; ++i)

            ar[i] = -1;

        a = ar;

    }

    public boolean del(int x) {

        for (int i = 0; i <= lastindex; ++i) {

            if (a[i] == x) {

                for (int j = i; j <= lastindex; ++j) {

                    a[j] = a[j + 1];

                }

                lastindex--;

                return true;

            }

        }

        return false;

    }

    public void display() {

        System.out.println("The array is " + Arrays.toString(a));

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        GSSArray ob;

        char ch;

```



```
System.out.print("Enter the size of array:");

int s = sc.nextInt();

ob = new GSSArray(s);

System.out.println("Menu");

System.out.println("1.Insert a value");

System.out.println("2.Delete a value");


do {

    System.out.print("Enter your choice:");

    int m = sc.nextInt();

    switch (m) {

        case 1:

            System.out.print("Enter the value:");

            int v = sc.nextInt();

            ob.insert(v);

            break;

        case 2:

            System.out.print("Enter the value to be deleted:");

            int d = sc.nextInt();

            if (ob.del(d)) {

                System.out.println("Deletion successful");

            } else {

                System.out.println("Element not found");

            }

            break;

        default:

            System.out.println("You have entered a wrong choice");

    }

}
```

```

    }

    ob.display();

    System.out.println("Do you want to continue(y/n):");

    ch = sc.next().charAt(0);

    } while (ch == 'y');

}
}

```

### Input/Output:

```

Enter the size of array:5
Menu
1.Insert a value
2.Delete a value
Enter your choice:1
Enter the value:3
The array is [3, -1, -1, -1, -1]
Do you want to continue(y/n):
y
Enter your choice:1
Enter the value:2
The array is [2, 3, -1, -1, -1]
Do you want to continue(y/n):
y
Enter your choice:1
Enter the value:5
The array is [2, 3, 5, -1, -1]
Do you want to continue(y/n):
n

```

**Conclusion:**

From this practical we learnt implementation of different array operations.

**Signature of Teacher:**