



Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

**Faculty of Sciences and Engineering, Semester: Spring, Year:
2024, B.Sc. in CSE (weekend)**

Lab Report # 02

Course Title: Object Oriented Programming Lab

Course Code: CSE-202

Section: 223 E1

Student Details

Name		ID
1.	Waliullah	223015026

Date : 08-03-2024

Submission Date : 15-03-2024

Course Teacher's Name : Abdullah Al Farhad

Assignment Status

Marks:

Signature:.....

Comments:.....

Date:.....

Title: Take three constructor where first constructor will calculate the area of triangle, second constructor will calculate the area of rectangle and third constructor will calculate the area of circle using overloading constructor. However, Input must be taken from users. Also using switch case.

1. Introduction:

- Constructor overloading is a powerful technique in object-oriented programming that allows a class to have multiple constructors with different parameter lists. This enables the creation of objects in various ways, providing flexibility for initialization based on the required data. In this lab, we explored constructor overloading to design a program that calculates the area of different shapes: triangle, rectangle, and circle.

2. Objective:

- To gain practical experience with constructor overloading in a class hierarchy.
- To implement user input mechanisms to obtain shape dimensions during object creation.
- To design separate constructors for each shape, handling data validation for meaningful area calculations.
- Design a class named **Area** to encapsulate functionalities related to area calculation.
- Implement constructors for the **Area** class to accommodate different shapes.
- Develop methods within the **Area** class to calculate the area for each shape.
- Create a program (**Lab2**) that interacts with the user to choose the desired shape and obtain necessary measurements.
- Calculate and display the area based on **user** input.

3. Experiment Setup:

1. Triangle Constructor:

- The first constructor is designed to calculate the area of a triangle.
- Users are prompted to input the base and height of the triangle.
- The constructor utilizes the formula: $\text{Area} = (a + b + c) / 2$ and $\text{result} = (\text{result} * (\text{result} - a) * (\text{result} - b) * (\text{result} - c))$.

2. Rectangle Constructor:

- The second constructor is responsible for computing the area of a rectangle.
- Users are required to enter the length and width of the rectangle.
- The constructor employs the formula: $\text{Area} = \text{Length} \times \text{Width}$.

3. Circle Constructor:

- The third constructor is designed for calculating the area of a circle.
- Users input the radius of the circle.
- The constructor uses the formula: $\text{Area} = \text{PI} \times \text{Radius} \times \text{Radius}$.

4. Procedure:

- The program prompts the user to choose the shape for which they want to calculate the area (triangle, rectangle, or circle).

- Depending on the user's choice, the corresponding constructor is invoked, and the user is prompted to enter the required parameters for that shape.
- The program then calculates and displays the area of the specified shape.
- Users can repeat the process for different shapes as needed.

5. Implementations and Output:

- Code:

Lab2.java ×

Lab_Report2 > LabReportCodes > Lab2.java > Area > Area(double)

```
1
2  import java.util.Scanner;
3
4  // area calculate class created
5  class Area {
6      double a, b, c, h, w, r, result;
7
8      // traingle calculate constructor created
9      Area(double a, double b, double c) {
10         this.a = a;
11         this.b = b;
12         this.c = c;
13     }
14
15     // rectangle calculate constructor created
16     Area(double h, double w) {
17         this.h = h;
18         this.w = w;
19     }
20
21     // circle calculate constructor created
22     Area(double r) {
23         this.r = r;
24     }
```

Lab2.java ×

Lab_Report2 > LabReportCodes > Lab2.java > Area > Area(double)

```
5  class Area {
25
26      // traingle calculate method created
27      double traingleCalculate() {
28          result = (a + b + c) / 2;
29          return Math.sqrt(result * (result - a) * (result - b) * (result - c));
30      }
31
32      // rectangle calculate method created
33      double rectangleCalculate() {
34          return h * w;
35      }
36
37      // circle calculate method created
38      double circleCalculate() {
39          return Math.PI * r * r;
40      }
41
42  }
43
```

Lab2.java ×

Lab_Report2 > LabReportCodes > Lab2.java > Area > Area(double)

```
44  public class Lab2 {
45      Run | Debug
46      public static void main(String[] args) {
47
48          Scanner input = new Scanner(System.in);
49          System.out.println();
50          System.out.print(s:"Enter Your Spacific Case (Like traingle, rectangle, circle): ");
51          String name = input.nextLine();
52
53          switch (name) {
54              case "traingle":
55                  // traingle value input
56                  System.out.println();
57                  System.out.println(x:"Traingle Value Input");
58                  System.out.print(s:"Input Value of a: ");
59                  int a = input.nextInt();
60                  System.out.print(s:"Input Value of b: ");
61                  int b = input.nextInt();
62                  System.out.print(s:"Input Value of c: ");
63                  int c = input.nextInt();
64                  Area trainangleObj = new Area(a, b, c);
65
66                  System.out.print(s:"Traingle Value is: ");
67                  System.out.println(trainangleObj.traingleCalculate());
68                  break;
69              case "rectangle":
70                  // rectangle value input
71                  System.out.println();
72                  System.out.println(x:"Rectangle Value Input");
73                  System.out.print(s:"Input Value of h: ");
74                  int h = input.nextInt();
75                  System.out.print(s:"Input Value of w: ");
76                  int w = input.nextInt();
77                  Area rectObj = new Area(h, w);
78                  System.out.print(s:"Rectangle Value is: ");
79                  System.out.println(rectObj.rectangleCalculate());
80                  break;
81              case "circle":
82                  // circle value input
83                  System.out.println();
84                  System.out.println(x:"Circle Value Input");
85                  System.out.print(s:"Input Value of r: ");
86                  int r = input.nextInt();
87                  Area circleObj = new Area(r);
88                  System.out.print(s:"Circle Value is: ");
89                  System.out.println(circleObj.circleCalculate());
90                  break;
91              default:
92                  System.out.println(s:"Invalid Case");
93          }
94      }
95  }
```

Lab2.java ×

Lab_Report2 > LabReportCodes > Lab2.java > Area > Area(double)

```
44 public class Lab2 {
45     public static void main(String[] args) {
67         break;
68         case "rectangle":
69             // rectangle value input
70             System.out.println();
71             System.out.println(x:"Rectangle Value Input");
72             System.out.print(s:"Input Value of Height: ");
73             int h = input.nextInt();
74             System.out.print(s:"Input Value of Width: ");
75             int w = input.nextInt();
76             Area rectangleObj = new Area(h, w);
77
78             System.out.print(s:"Rectangle Value is: ");
79             System.out.println(rectangleObj.rectangleCalculate());
80             break;
81         case "circle":
82             // circle value input
83             System.out.println();
84             System.out.println(x:"Circle Value Input");
85             System.out.print(s:"Input Value of Radius: ");
86             int r = input.nextInt();
87             Area circleObj = new Area(r);
88
89             System.out.print(s:"Circle Value is: ");
90             System.out.println(circleObj.circleCalculate());
```

Lab2.java ×

Lab_Report2 > LabReportCodes > Lab2.java > Area > Area(double)

```
44 public class Lab2 {
45     public static void main(String[] args) {
89         System.out.print(s:"Circle Value is: ");
90         System.out.println(circleObj.circleCalculate());
91         break;
92
93         default:
94             System.out.println(
95                 x:"This item is not found. Please try again and Exact value. You can write input line
                    carefully!");
96             break;
97     }
98     input.close();
99 }
100 }
```

- Output:

```
Waliullah@Waliullah MINGW64 /d/OOP Lab University (main)
$ cd "/d/OOP Lab University/Lab_Report2/LabReportCodes/" && javac Lab2.java && java Lab2

Enter Your Spacific Case (Like traingle, rectangle, circle): traingle

Traingle Value Input
Input Value of a: 5
Input Value of b: 6
Input Value of c: 9
Traingle Value is: 14.142135623730951

Waliullah@Waliullah MINGW64 /d/OOP Lab University/Lab_Report2/LabReportCodes (main)
$
```

```
Waliullah@Waliullah MINGW64 /d/OOP Lab University (main)
$ cd "/d/OOP Lab University/Lab_Report2/LabReportCodes/" && javac Lab2.java && java Lab2

Enter Your Spacific Case (Like traingle, rectangle, circle): rectangle

Rectangle Value Input
Input Value of Height: 5
Input Value of Width: 6
Rectangle Value is: 30.0

Waliullah@Waliullah MINGW64 /d/OOP Lab University/Lab_Report2/LabReportCodes (main)
$
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Waliullah@Waliullah MINGW64 /d/OOP Lab University (main)

\$ cd "/d/OOP Lab University/Lab_Report2/LabReportCodes/" && javac Lab2.java && java Lab2

Enter Your Spacific Case (Like traingle, rectangle, circle): circle

Circle Value Input

Input Value of Radius: 9

Circle Value is: 254.46900494077323

Waliullah@Waliullah MINGW64 /d/OOP Lab University/Lab_Report2/LabReportCodes (main)

\$

public class Lab2 {

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Waliullah@Waliullah MINGW64 /d/OOP Lab University (main)

\$ cd "/d/OOP Lab University/Lab_Report2/LabReportCodes/" && javac Lab2.java && java Lab2

Enter Your Spacific Case (Like traingle, rectangle, circle): adaidf

This item is not found. Please try again and Exact value. You can write input line carefully!

Waliullah@Waliullah MINGW64 /d/OOP Lab University/Lab_Report2/LabReportCodes (main)

\$

6. Limitations:

- The input for dimensions (e.g., sides of the triangle, height and width of the rectangle, radius of the circle) is expected to be integers. Consider using **double** instead of **int** to handle decimal values for more accurate calculations.
- The code doesn't handle negative input values. You might want to include input validation to ensure positive values.
- The code could benefit from more comments explaining the logic of the calculations and any user prompts.
- Closing the **Scanner** at the end of the program is good practice, but it might be better to use a try-with-resources statement to ensure proper resource management.

7. Conclusion:

- This lab successfully demonstrated the application of constructor overloading for creating objects of different shapes with user-provided dimensions. The implemented input validation ensured meaningful area calculations. By designing separate constructors and area calculation methods for each shape, the code achieved clarity and maintainability.

