

# Rajalakshmi Engineering College

Name: Kaaviya Sri PS  
Email: 240701222@rajalakshmi.edu.in  
Roll no: 240701222  
Phone: 8838174850  
Branch: REC  
Department: CSE - Section 6  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## 2024\_28\_III\_OOPS Using Java Lab

### 2028\_REC\_OOPS using Java\_Week 2\_CY

Attempt : 1  
Total Mark : 40  
Marks Obtained : 40

#### Section 1 : Coding

##### 1. Problem Statement

Raj is solving a physics problem involving projectile motion, where he needs to calculate the time a ball hits the ground using a quadratic equation of the form  $ax^2 + bx + c = 0$ . Depending on the coefficients, the ball may hit the ground once, twice, or not at all in real time.

Help Raj find all real roots of the equation, if any.

Note: discriminant =  $b^2 - 4ac$

##### ***Input Format***

The input consists of three space-separated doubles a, b, and c, representing the coefficients of the quadratic equation.

##### ***Output Format***

If there are two real roots, print:

- "Two real solutions:"
- "Root1 = <value>"
- "Root2 = <value>"

If there is one real root, print:

- "One real solution:"
- "Root = <value>"

If there are no real roots, print:

- "There are no real solutions."

Note: values are rounded to two decimal places.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 1 6 9

Output: One real solution:

Root = -3.00

### **Answer**

```
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double a = sc.nextDouble();
        double b = sc.nextDouble();
        double c = sc.nextDouble();
        double discriminant = b * b - 4 * a * c;

        if (discriminant > 0) {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("Two real solutions:");
```

```

        System.out.printf("Root1 = %.2f\n", root1);
        System.out.printf("Root2 = %.2f\n", root2);
    } else if (discriminant == 0) {
        double root = -b / (2 * a);
        System.out.println("One real solution:");
        System.out.printf("Root = %.2f\n", root);
    } else {
        System.out.println("There are no real solutions.");
    }
}
}
}

```

**Status :** Correct

**Marks :** 10/10

## 2. Problem Statement

Samantha is a diligent math student who is exploring the world of programming. She is learning Java and has recently studied conditional statements. One day, her teacher gives her an interesting problem to solve, which takes a number as input and checks whether it is a multiple of 5 or 7.

Help her complete the task.

### **Input Format**

The input consists of a single integer N, representing the number to be checked.

### **Output Format**

If the number is a multiple of 5 but not 7, the output prints "N is a multiple of 5".

If the number is a multiple of 7, the output prints "N is a multiple of 7".

Otherwise the output prints "N is neither multiple of 5 nor 7" where N is an entered integer.

Refer to the sample output for formatting specifications.

### Sample Test Case

Input: 10

Output: 10 is a multiple of 5

### Answer

```
import java.util.*;

class Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();

        if (N % 5 == 0 && N % 7 != 0) {
            System.out.println(N + " is a multiple of 5");
        } else if (N % 7 == 0) {
            System.out.println(N + " is a multiple of 7");
        } else {
            System.out.println(N + " is neither multiple of 5 nor 7");
        }
    }
}
```

Status : Correct

Marks : 10/10

### 3. Problem Statement

John is a fitness trainer, and he wants to use the BMI calculator to assess the body mass index of his clients. He has a list of clients based on their height and weight.

John plans to write a program to quickly determine the BMI and provide a classification for each client.

If BMI is less than 18.5, the program will classify it as "Underweight" If BMI is between 18.6 and 24.9, the program will classify it as "Normal Weight" If BMI is between 25.0 and 29.9, the program will classify it as "Overweight" If BMI is 30.0 or higher, the program will classify it as "Obese"

Note: Formula to calculate BMI =  $\text{weight}/(\text{height} \times \text{height})$

### ***Input Format***

The first line of input consists of a double value, representing the height of the person in meters.

The second line consists of a double value, representing the weight of the person in kilograms.

### ***Output Format***

The first line of output prints "BMI: " followed by a double (rounded to two decimal places) representing the calculated BMI.

The second line prints "Classification: " followed by a string indicating the BMI category (Underweight, Normal Weight, Overweight, or Obese).

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 1.2

45.2

Output: BMI: 31.39

Classification: Obese

### ***Answer***

```
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        double height = sc.nextDouble();
        double weight = sc.nextDouble();

        double bmi = weight / (height * height);

        System.out.printf("BMI: %.2f\n", bmi);

        if (bmi < 18.5) {
            System.out.println("Classification: Underweight");
        }
    }
}
```

```
} else if (bmi >= 18.6 && bmi <= 24.9) {  
    System.out.println("Classification: Normal Weight");  
} else if (bmi >= 25.0 && bmi <= 29.9) {  
    System.out.println("Classification: Overweight");  
} else {  
    System.out.println("Classification: Obese");  
}  
  
    sc.close();  
}  
}
```

**Status :** Correct

**Marks : 10/10**

#### 4. Problem Statement

Maya, a student in an arts and crafts class, wants to create a pattern using stars (\*) in a specific format. She plans to use a program to help her construct the pattern.

Write a program that takes an integer as input and constructs the following pattern using nested for loops.

Input: 5

Output:

```
*  
* *  
* * *  
* * * *  
* * * * *  
* * * *  
* * *  
* *  
*  

```

### ***Input Format***

The input consists of a number (integer) representing the number of rows.

### ***Output Format***

The output displays the required pattern.

Refer to the sample output for the formatting specifications.

### ***Sample Test Case***

Input: 5

Output: \*

```
* *
* * *
* * * *
* * * * *
* * * *
* * * *
* * *
* *
*
```

### ***Answer***

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();

        for (int i = 1; i <= N; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }

        for (int i = N-1; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print("* ");
            }
        }
    }
}
```

```
240701222 System.out.println();  
240701222 }  
240701222 }  
240701222 }
```

**Status :** Correct

**Marks :** 10/10