

# Rajalakshmi Engineering College

Name: SANTHOSH S  
Email: 241801251@rajalakshmi.edu.in  
Roll no: 241801251  
Phone: 9790911586  
Branch: REC  
Department: AI & DS - Section 3  
Batch: 2028  
Degree: B.E - AI & DS

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

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***

```
// You are using Java
import java.util.Scanner;
class StarPattern {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
```

```

int rows = scanner.nextInt();

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

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

scanner.close();
}
}

```

**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**

```
// You are using Java
import java.util.Scanner;
class MultipleChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int N = scanner.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");
        }

        scanner.close();
    }
}
```

**Status :** Correct

**Marks :** 10/10

### **3. Problem Statement**

Ted, the computer science enthusiast, has accepted the challenge of writing a program that checks if the number of digits in an integer matches

the sum of its digits.

Guide Ted in designing and writing the code to solve this problem using a 'do-while' loop.

### ***Input Format***

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

### ***Output Format***

If the sum is equal to the number of digits, print "The number of digits in N matches the sum of its digits."

Else, print "The number of digits in N does not match the sum of its digits."

Refer to the sample output for formatting specifications.

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

Input: 20

Output: The number of digits in 20 matches the sum of its digits.

### ***Answer***

```
// You are using Java
import java.util.Scanner;
class DigitSumChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
```

```
        int N = scanner.nextInt();
        int temp = N;
        int sum = 0;
        int count = 0;
```

```
        do {
            int digit = temp % 10;
            sum += digit;
            count++;
            temp /= 10;
        } while (temp > 0);
```

```

        if (sum == count) {
            System.out.println("The number of digits in " + N + " matches the sum of
its digits.");
        } else {
            System.out.println("The number of digits in " + N + " does not match the
sum of its digits.");
        }

        scanner.close();
    }
}

```

**Status :** Correct

**Marks : 10/10**

#### 4. Problem Statement

Ram wants to evaluate the time required to break even on an investment based on initial costs, monthly profits, and monthly expenses. Write a program to calculate the break-even point in months and categorize the return on investment.

Compute the break-even point by using the formula:  $\text{initial cost} / (\text{monthly profit} - \text{monthly expenses})$ . Based on the break-even point, classify the return on investment into one of the following categories: Quick Return: If the break-even point is 3 months or fewer. Average Return: If the break-even point is between 4 and 12 months, inclusive. Long-term Return: If the break-even point exceeds 12 months.

Ram is new to programming, so he seeks your assistance in creating the program.

Note: monthly profit is always greater than monthly expenses.

#### **Input Format**

The first line of input consists of a double value representing the initial cost.

The second line consists of a double value representing the monthly profit.

The third line consists of a double value representing the monthly expenses.

### **Output Format**

The first line prints "Break-even Point:", followed by the break-even point as a decimal number (of double datatype), formatted to two decimal places.

The second line prints "Category: ", followed by the investment return as a String, which can be one of:

- "Quick Return" if break-even point  $\leq 3$
- "Average Return" if break-even point  $\leq 12$
- "Long-term Return" if break-even point  $> 12$

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 10000.50  
5000.75  
1000.10

Output: Break-even Point: 2.50  
Category: Quick Return

### **Answer**

```
// You are using Java
import java.util.Scanner;
class InvestmentAnalyzer {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        double initialCost = scanner.nextDouble();
        double monthlyProfit = scanner.nextDouble();
        double monthlyExpenses = scanner.nextDouble();

        double breakEvenPoint = initialCost / (monthlyProfit - monthlyExpenses);

        System.out.printf("Break-even Point: %.2f\n", breakEvenPoint);

        String category;
        if (breakEvenPoint <= 3) {
            category = "Quick Return";
```

```
    } else if (breakEvenPoint <= 12) {  
        category = "Average Return";  
    } else {  
        category = "Long-term Return";  
    }  
  
    System.out.println("Category: " + category);  
  
    scanner.close();  
    }  
}
```

**Status :** Correct

**Marks :** 10/10