RAJALAKSHMI ENGINEERING COLLEGE

(An Autonomous Institution)

RAJALAKSHMI NAGAR, THANDALAM- 602 105

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE

CS23333 – Object Oriented Programming Using JAVA

LABORATORY RECORD NOTEBOOK

NAME: K S VISHRUTHI

YEAR/SEMESTER: II – III

BRANCH/SECTION: BTECH – ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

REGISTER NO: 241801318

COLLEGE ROLL NO: 2116 241801318

ACADEMIC YEAR: 2025-2026

RAJALAKSHMI ENGINEERING COLLEGE

(An Autonomous Institution)

RAJALAKSHMI NAGAR, THANDALAM- 602 105

BONAFIDE CERTIFICATE

NAME: K S VISHRUTHI **BRANCH/SECTION:** B.TECH - ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

ACADEMIC YEAR: 2025-2026 **SEMESTER:** III

REGISTER NO: 2116241801318

Certified that this is a Bonafide record of work done by the above student in the **CS23333 - Object Oriented Programming Using JAVA** during the year 20 - 20

Signature of Faculty In-charge

Submitted for the Practical Examination Held on: 22/11/2025

Internal Examiner **External Examiner**

INDEX

Modules	EXERCISE TITLE	Date	PAGE NO.	Sign
1.	Week 1 - I/O Data Types Operators	24-07-2025		
2.	Week 2 - Control Structures	01-08-2025		
3.	Week 3 - Arrays	14-08-2025		
4.	Week 4 - Strings	29-08-2025		
5.	Week 5 - Classes & Objects	12-09-2025		
6.	Week 6 - Inheritance	25-09-2025		
7.	Week 7 - Interface	03-10-2025		
8.	Week 8 - Exceptions	17-10-2025		
9.	Week 9 – Collections	30-10-2025		
10.	Week 10 – Collections	07-11-2025		
11.	Week 11 – Project	14-11-2025		
12.	Week 12 - Lambda	20-11-2025		

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Gloria is responsible for monitoring the performance of two machines in a factory. She needs to determine which of the two machines is operating closest to the optimal temperature of 100 degrees Celsius using the relational operator.

Assist Gloria in displaying the machine's temperature, which is closer to 100, and the difference from 100.

Input Format

The first line of input consists of an integer N, representing the temperature of the first machine.

The second line consists of an integer M, representing the temperature of the second machine.

Output Format

The output prints "The integer closer to 100 is X with a difference of Y" where X is the temperature of the closer machine and Y is the difference from 100.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 90
80

Output: The integer closer to 100 is 90 with a difference of 10

Answer

```
import java.util.Scanner;
class main{
   public static void main(String[] args){
      Scanner sc=new Scanner(System.in);
      int temp1=sc.nextInt();
      int temp2=sc.nextInt();
      int diff1=Math.abs(temp1-100);
      int diff2=Math.abs(temp2-100);
      if(diff1<diff2){
         System.out.println("The integer closer to 100 is "+temp1+" with a
difference of "+diff1);
      }else{
         System.out.println("The integer closer to 100 is "+temp2+" with a
difference of "+diff2);
      }
      sc.close();
   }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. PROBLEM STATEMENT:

Dave got two students who wants help with their doubt. Each handouts an integer and wants to find if one Integer Positive While the Other is Not Divisible by 3. Write a program to achieve this and conclude for them.

Input Format

The first line of input represents the first integer.

The second line of input represents the second integer.

Output Format

The output should display as "One of the integers is positive while the other is not divisible by 3." or "Neither of the integers meets the condition."

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 4
3
Output: One of the integers is positive while the other is not divisible by 3.

Answer

```
// You are using Java
import java.util.*;
class main{
   public static void main(String[] args){
      Scanner sc =new Scanner(System.in);
      int a=sc.nextInt();
      int b=sc.nextInt();
      if((a>0&&b%3!=0)||(b>0&&a%3!=0)){
         System.out.println("One of the integers is positive while the other is not
divisible by 3.");
      }
      else{
         System.out.println("Neither of the integers meets the condition.");
      }
   }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
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_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Arun is working on a project to automate the process of determining whether a student has passed or failed based on their subject marks.

He aims to create a simple program that takes positive integers as marks for five subjects from the user. If the average of the marks is greater than or equal to 50, the student has passed the exam. Otherwise, the student has failed.

Help Arun to implement the project.

Input Format

The input consists of five space-separated integers, representing the marks in five subjects.

Output Format

The first line of output prints "Average score: " followed by an integer representing the average score.

The second line prints one of the following:

1. If the condition is satisfied, print "The student has passed".
2. Otherwise, the output prints "The student has failed".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50 60 70 80 90

Output: Average score: 70
The student has passed

Answer

```
// You are using Java
import java.util.*;
class main{
   public static void main(String[] args){
      Scanner sc=new Scanner(System.in);
      int a=sc.nextInt();
      int b=sc.nextInt();
      int c=sc.nextInt();
      int d=sc.nextInt();
      int e=sc.nextInt();
      int avg=(a+b+c+d+e)/5;
      if(avg>=50){
         System.out.printf("Average score: %d\n The student has passed",avg);
      }else{
         System.out.printf("Average score: %d\n The student has failed",avg);
      }
   }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
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_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. 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.*;
class main{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        if(n%5==0){
            System.out.printf("%d is a multiple of 5",n);
        }else if(n%7==0){
            System.out.printf("%d is a multiple of 7",n);
        }else{
            System.out.printf("%d is neither multiple of 5 nor 7",n);
        }
    }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 3_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Rosh is intrigued by numerical patterns. Today, she stumbled upon a puzzle while working with arrays. She wants to compute the sum of the third-largest and second-smallest elements from a list of integers. She seeks your help to implement a program that solves this for her efficiently.

Input Format

The first line of input is an integer N, representing the size of the array.

The second line of input consists of N space-separated integers, representing the elements of the array.

Output Format

The output displays a single integer representing the sum of the third-largest and second-smallest elements in the array.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10
10 20 30 40 50 60 70 80 90 100

Output: 100

Answer

```
// You are using Java
import java.util.*;
class main{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[] arr=new int[n];
        for(int i=0;i<n;i++){
            arr[i]=sc.nextInt();
        }
        Arrays.sort(arr);
        int thirdlar=arr[n-3];
        int secondsma=arr[1];
        System.out.println(thirdlar+secondsma);
    }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 3_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Monica is interested in finding a treasure but the key to opening is to get the sum of the main diagonal elements and secondary diagonal elements.

Write a program to help Monica find the diagonal sum of a square 2D array.

Note: The main diagonal of the array consists of the elements traversing from the top-left corner to the bottom-right corner. The secondary diagonal includes elements from the top-right corner to the bottom-left corner.

Input Format

The first line of input consists of an integer N, representing the number of rows and columns.

The following N lines consist of N space-separated integers, representing the 2D array elements.

Output Format

The first line of output prints "Sum of the main diagonal: " followed by an integer, representing the sum of the main diagonal.

The second line prints "Sum of the secondary diagonal: " followed by an integer, representing the sum of the secondary diagonal.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3
1 2 3
4 5 6
7 8 9

Output: Sum of the main diagonal: 15
Sum of the secondary diagonal: 15

Answer

```
// You are using Java
import java.util.*;
class main{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int[][] arr=new int[n][n];
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                arr[i][j]=sc.nextInt();
            }
        }
        int mainds=0;
        int secds=0;
        for(int i=0;i<n;i++){
            mainds+=arr[i][i];
            secds+=arr[i][n-1-i];
        }
```

```
        System.out.println("Sum of the main diagonal:"+mainds);
        System.out.println("Sum of the secondary diagonal:"+secds);
    }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 4_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In a publishing company, editors often need to quickly analyze passages of text to check for punctuation usage. To assist them, you are asked to write a program that counts the number of specific punctuation marks in each passage.

The punctuation marks of interest are:

Commas (,)Periods (.)Question marks (?)

Input Format

The first line of input contains an integer T, representing the number of test cases (passages).

Each of the next T lines contains a single passage of text.

Output Format

For each test case, print three integers separated by spaces, representing the number of commas, periods, and question marks in the passage.

The first line of output corresponds to the first passage, the second line to the second passage, and so on.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
Hello, world. How are you?
Output: 1 1 1

Answer

```
// You are using Java
import java.util.*;
class main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int T = Integer.parseInt(sc.nextLine());

        for (int t = 0; t < T; t++) {
            String passage = sc.nextLine();
            int commas = 0, periods = 0, questions = 0;

            for (char ch : passage.toCharArray()) {
                if (ch == ',') commas++;
                else if (ch == '.') periods++;
                else if (ch == '?') questions++;
            }

            System.out.println(commas + " " + periods + " " + questions);
        }

        sc.close();
    }
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 4_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Anu is developing a tool for a conference registration system. Participants submit keywords related to their fields of interest. The organizer wants to sort these keywords alphabetically to generate tags for session grouping.

Write a program that accepts at least five keywords as input arguments and outputs them in sorted alphabetical order.

Input Format

The first line of input contains an integer n, representing the number of keywords.

The second line of input contains n space-separated keywords (string).

Output Format

The output prints n space separated strings representing the sorted keyword in alphabetical order.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
Blockchain Cloud AI Data Cybersecurity
Output: AI Blockchain Cloud Cybersecurity Data

Answer

```
// You are using Java
import java.util.*;
class main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        String[] keywords = new String[n];
        for (int i = 0; i < n; i++) {
            keywords[i] = sc.next();
        }
        Arrays.sort(keywords);
        for (int i = 0; i < n; i++) {
            System.out.print(keywords[i]);
            if (i < n - 1) System.out.print(" ");
        }
    }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are working as a developer for CityBank, which wants to build a basic account management system.

Each customer at the bank has:

An Account Number (integer)A Customer Name (string)An Initial Balance (double)

The bank allows two types of transactions:

Deposit – increases the balance.Withdrawal – decreases the balance only if enough funds are available.

If the withdrawal amount is greater than the balance, the withdrawal should not happen, and the balance should remain the same.

You are required to implement this system using:

A class with attributes for account details.A constructor to initialize account details.Setter methods to update details if needed.Getter methods to retrieve details.Objects of the class to represent customers.

Finally, display each customer’s account details after all transactions.

Input Format

The first line of input contains an integer N, representing the number of customers.

For each customer:

- The next line contains the account number (integer).
- The following line contains the customer name (string).
- The next line contains the initial balance (double).
- The next line contains the deposit amount (double).
- The next line contains the withdrawal amount (double).

Output Format

For each customer, print the details in the following format:

1. Account Number: <account_number>
2. Customer Name: <customer_name>
3. Final Balance: <final_balance> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
1234
Rahul Sharma
5000
2000
3000
Output: Account Number: 1234
Customer Name: Rahul Sharma

Final Balance: 4000.0

Answer

```
// You are using Java
import java.util.*;
class acct{
    private int acctnum;
    private String custname;
    private double balance;
    public acct(int acctnum,String custname,double balance){
        this.acctnum=acctnum;
        this.custname=custname;
        this.balance=balance;
    }
    public void deposit(double amount){
        if(amount>0){
            balance+=amount;
        }
    }
    public void withdraw(double amount){
        if(amount>0&&amount<=balance){
            balance-=amount;

        }
    }
    public void display(){
        System.out.println("Account Number:"+acctnum);
        System.out.println("Customer Name:"+custname);
        System.out.println("Final Balance:"+balance);
    }
}
class main{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        for(int i=0;i<n;i++){
            int accno=sc.nextInt();
            sc.nextLine();
            String name=sc.nextLine();
            double initialbalance=sc.nextDouble();
            double depositamt=sc.nextDouble();
            double withdrawamt=sc.nextDouble();
```

```
            acct acc=new acct(accno,name,initialbalance);
            acc.deposit(depositamt);
            acc.withdraw(withdrawamt);
            acc.display();
        }
        sc.close();
    }
}
```

Status : Correct **Marks : 10/10**

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Neha is working as a developer for CityElectricity Board, which wants to build a household electricity billing system.

Each customer's electricity account has:

A Customer ID (integer)A Customer Name (string)Units Consumed (double)

The electricity bill is calculated based on these rules:

For the first 100 units 5 units charge per unitFor the next 100 units (101–200) 7 units charge per unitFor units above 200 10 units charge per unitIf the total bill exceeds 2000 units, a 5% discount is applied on the final bill.

Neha has been asked to implement this system using:

A class with attributes for customer details.A constructor to initialize customer details.Setter methods to update details if needed.Getter methods to retrieve details.Objects of the class to represent customers.

Finally, display each customer's details and final bill amount.

Input Format

The first line of input contains an integer N, representing the number of customers.

For each customer:

- The next line contains the Customer ID (integer).
- The following line contains the Customer Name (string).
- The next line contains the Units Consumed (double).

Output Format

For each customer, print the details in the following format:

Customer ID: <customer_id>

Customer Name: <customer_name>

Final Bill: <final_bill> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
1001
Ravi Kumar
80
Output: Customer ID: 1001
Customer Name: Ravi Kumar
Final Bill: 400.0

Answer

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

```
class Customer {
    private int customerId;
    private String customerName;
    private double unitsConsumed;

    public Customer(int customerId, String customerName, double
unitsConsumed) {
        this.customerId = customerId;
        this.customerName = customerName;
        this.unitsConsumed = unitsConsumed;
    }

    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }

    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }

    public void setUnitsConsumed(double unitsConsumed) {
        this.unitsConsumed = unitsConsumed;
    }

    public int getCustomerId() {
        return customerId;
    }

    public String getCustomerName() {
        return customerName;
    }

    public double getUnitsConsumed() {
        return unitsConsumed;
    }

    public double calculateBill() {
        double bill = 0;

        if (unitsConsumed <= 100) {
            bill = unitsConsumed * 5;
```

```
        } else if (unitsConsumed <= 200) {
            bill = (100 * 5) + ((unitsConsumed - 100) * 7);
        } else {
            bill = (100 * 5) + (100 * 7) + ((unitsConsumed - 200) * 10);
        }

        if (bill > 2000) {
            bill = bill - (bill * 0.05);
        }

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

        int n = Integer.parseInt(sc.nextLine());

        for (int i = 0; i < n; i++) {
            int id = Integer.parseInt(sc.nextLine());
            String name = sc.nextLine();
            double units = Double.parseDouble(sc.nextLine());

            Customer c = new Customer(id, name, units);

            System.out.println("Customer ID: " + c.getCustomerId());
            System.out.println("Customer Name: " + c.getCustomerName());
            System.out.printf("Final Bill: %.1f\n", c.calculateBill());
        }

    }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Elsa subscribes to a premium service with a base monthly cost, a service tax and an extra feature cost. Assist her in writing an inheritance program that takes input for these values and calculates the total monthly cost.

Refer to the below class diagram:

Input Format

The first line of input consists of a double value, representing the base monthly cost.

The second line consists of a double value, representing the service tax.

The third line consists of a double value, representing the extra feature cost.

Output Format

The output prints "Rs. X" where X is a double value, rounded off to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 10.0
2.5
5.0
Output: Rs. 17.50

Answer

```
import java.util.Scanner;

// Parent class
class Subscription {
    protected double monthlyCost;
    protected double serviceTax;
    protected double extraFeatureCost;
}

class PremiumSubscription extends Subscription {

    public PremiumSubscription(double monthlyCost, double serviceTax, double
extraFeatureCost) {
        this.monthlyCost = monthlyCost;
        this.serviceTax = serviceTax;
        this.extraFeatureCost = extraFeatureCost;
    }

    public double calculateMonthlyCost() {
        return monthlyCost + serviceTax + extraFeatureCost;
    }
}

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

```
        double baseMonthlyCost = scanner.nextDouble();
        double serviceTax = scanner.nextDouble();
        double extraFeatureCost = scanner.nextDouble();

        PremiumSubscription premiumSubscription = new
PremiumSubscription(baseMonthlyCost, serviceTax, extraFeatureCost);

        double totalMonthlyCost = premiumSubscription.calculateMonthlyCost();

        System.out.printf("Rs. %.2f%n", totalMonthlyCost);

        scanner.close();
    }
}
```

Status : Correct **Marks : 10/10**

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Alice is managing an online store and wants to implement a program using inheritance to calculate the selling price of products after applying discounts.

Guide her by following the instructions:

Create a base class called Product with a public double attribute price.Create a subclass called DiscountedProduct, which extends Product and includes a private double attribute discount rate. This subclass has a method called calculateSellingPrice() to determine the final selling price after applying the discount.

Formula: Discounted selling price = price * (1 - discount rate)

Input Format

The first line of input consists of a double value p, the initial price of the product.

The second line consists of a double value d, the discount rate.

Output Format

The output prints "Rs. X", where X is a double value, representing the calculated discounted selling price, rounded off to two decimal places.

If the discount rate is greater than 1, print "Not applicable".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50.00
0.20

Output: Rs. 40.00

Answer

```
import java.util.Scanner;

// You are using Java
class Product {
    public double price;
}
class DiscountedProduct extends Product {
    private double discountRate;

    public DiscountedProduct(double price, double discountRate) {
        this.price = price;
        this.discountRate = discountRate;
    }

    public double calculateSellingPrice() {
        return price * (1 - discountRate);
    }
}

class ProductPricing {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
```

```
        double initialPrice = scanner.nextDouble();
        double discountRate = scanner.nextDouble();
        DiscountedProduct discountedProduct = new
DiscountedProduct(initialPrice, discountRate);
        double sellingPrice = discountedProduct.calculateSellingPrice();

        if (sellingPrice >= 0) {
            System.out.printf("Rs. %.2f%n", sellingPrice);
        } else {
            System.out.println("Not applicable");
        }
        scanner.close();
    }
}
```

Status : Correct **Marks : 10/10**

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 6.5

Section 1 : Coding

1. Problem Statement:

Rajiv is analyzing the energy consumption in his household and wants to calculate the total cost based on the daily energy usage. He is given the rate per unit of electricity and the energy consumed for multiple days. To structure this calculation efficiently, he decides to use an interface-based approach.

Implement an interface CostCalculator with the necessary methods to retrieve energy details and compute the cost. The calculations should be handled in the EnergyConsumptionTracker class, while the EnergyConsumptionApp class should only handle input and output.

Formula

Energy Cost for one day = Energy Consumed per day * Rate Per Unit

Input Format

The first line of input consists of the rate per unit as an 'R' (a double value).

The second line of input consists of the number of days 'N' (an integer).

The third line of input consists of the daily energy consumption values for each day 'D" (double values), separated by space.

Output Format

The first line of the output prints: "Day-wise Energy Cost:"

The next N lines of the output print the day-wise energy costs(double type) and the total energy cost (double type) in Indian Rupees in the following format: "Day [day_number]: Rs. [energy_cost]"

The last line of the output prints: "Total Energy Cost: Rs. [total_cost]"

Note: energy_cost and total_cost are rounded off to two decimal points

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 0.01
3
10.0 20.0 30.0
Output: Day-wise Energy Cost:
Day 1: Rs. 0.10
Day 2: Rs. 0.20
Day 3: Rs. 0.30
Total Energy Cost: Rs. 0.60

Answer

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

```
interface CostCalculator {
    void getEnergyDetails(Scanner scanner);
    void calculateAndDisplayCost();
}

class EnergyConsumptionTracker implements CostCalculator {
    private double ratePerUnit;
    private int numDays;
    private double[] dailyConsumption;

    public EnergyConsumptionTracker(double ratePerUnit, int numDays) {
        this.ratePerUnit = ratePerUnit;
        this.numDays = numDays;
        this.dailyConsumption = new double[numDays];
    }

    @Override
    public void getEnergyDetails(Scanner scanner) {
        String line = scanner.nextLine().trim();
        while (line.isEmpty()) {
            line = scanner.nextLine().trim();
        }
        String[] values = line.split(" ");
        for (int i = 0; i < numDays; i++) {
            dailyConsumption[i] = Double.parseDouble(values[i]);
        }
    }

    @Override
    public void calculateAndDisplayCost() {
        System.out.println("Day-wise Energy Cost:");
        double totalCost = 0.0;
        for (int i = 0; i < numDays; i++) {
            double cost = dailyConsumption[i] * ratePerUnit;
            totalCost += cost;
            System.out.printf("Day %d: Rs. %.2f%n", i + 1, cost);
        }
        System.out.printf("Total Energy Cost: Rs. %.2f%n", totalCost);
    }
}
```

```
class EnergyConsumptionApp {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        double ratePerUnit = scanner.nextDouble();
        int numDays = scanner.nextInt();

        CostCalculator tracker = new EnergyConsumptionTracker(ratePerUnit,
numDays);

        tracker.getEnergyDetails(scanner);
        tracker.calculateAndDisplayCost();

        scanner.close();
    }
}
```

Status : Partially correct **Marks : 6.5/10**

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Jaheer is working on a health monitoring system to help individuals calculate their Body Mass Index (BMI). He has implemented a basic BMI calculator and an interface called HealthCalculator. It should have a method called calculateBMI.

You are tasked with creating a program that takes weight and height as input, calculates the BMI using the BMICalculator class, and displays the result. If the height or weight is less than or equal to zero, then return -1.

Formula: BMI = weight / (height * height)

Input Format

The first line of input consists of a double value W, the person's weight in kilograms.

The second line consists of a double value H, the height of the person in meters.

Output Format

The output displays "BMI: " followed by a double value, representing the calculated BMI, rounded off to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 70.0
1.75
Output: BMI: 22.86

Answer

```
import java.util.Scanner;


interface HealthCalculator {
    double calculateBMI(double weight, double height);
}

class BMICalculator implements HealthCalculator {
    @Override
    public double calculateBMI(double weight, double height) {
        if (weight <= 0 || height <= 0) {
            return -1;
        }
        return weight / (height * height);
    }
}



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

        double weight = scanner.nextDouble();
        double height = scanner.nextDouble();
```

```
        BMICalculator bmiCalculator = new BMICalculator();
        double bmi = bmiCalculator.calculateBMI(weight, height);
        System.out.printf("BMI: %.2f\n", bmi);

        scanner.close();
    }
}
```

Status : Correct **Marks : 10/10**

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Write a program to validate the email address and display suitable exceptions if there is any mistake.

Create 3 custom exception classes as below

DotExceptionAtTheRateExceptionDomainException

A typical email address should have a " . " character, and a "@" character, and also the domain name should be valid. Valid domain names for practice be 'in', 'com', 'net', or 'biz'.

Display Invalid Dot usage, Invalid @ usage, or Invalid Domain message based on email id.

Get the email address from the user, validate the email by checking the

above-mentioned criteria, and print the validity status of the input email address.

Input Format

The first line of input contains the email to be validated.

Output Format

The output prints a Valid email address or an Invalid email address along with the suitable exception

If email ends with . or contains not exactly one . after @, it throws:

DotException: Invalid Dot usage

Invalid email address

If @ appears not exactly once, it throws:

AtTheRateException: Invalid @ usage

Invalid email address

If the part after the last dot is not among accepted domains:

DomainException: Invalid Domain

Invalid email address

If all conditions satisfied then print:

Valid email address

Refer to the sample input and output for format specifications.

Sample Test Case

Input: sample@gmail.com

Output: Valid email address

Answer

```
import java.util.*;

class DotException extends Exception {
    DotException(String msg) {
        super(msg);
    }
}

class AtTheRateException extends Exception {
    AtTheRateException(String msg) {
        super(msg);
    }
}

class DomainException extends Exception {
    DomainException(String msg) {
        super(msg);
    }
}

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

        try {

            validateEmail(email);
            System.out.println("Valid email address");
        }
        catch (DotException e) {
            System.out.println("DotException: " + e.getMessage());
```

```
            System.out.println("Invalid email address");
        }
        catch (AtTheRateException e) {
            System.out.println("AtTheRateException: " + e.getMessage());
            System.out.println("Invalid email address");
        }
        catch (DomainException e) {
            System.out.println("DomainException: " + e.getMessage());
            System.out.println("Invalid email address");
        }
    }


    public static void validateEmail(String email) throws DotException,
AtTheRateException, DomainException {

        if (email.startsWith(".") || email.startsWith("@") || email.endsWith(".") ||
email.endsWith("@")) {
            throw new DotException("Invalid Dot usage");
        }


        int atCount = 0;
        for (char c : email.toCharArray()) {
            if (c == '@') atCount++;
        }
        if (atCount != 1) {
            throw new AtTheRateException("Invalid @ usage");
        }

        if (email.contains("..") || email.contains("@@")) {
            throw new DotException("Invalid Dot usage");
        }

        int atIndex = email.indexOf('@');
        int dotIndex = email.indexOf('.', atIndex);
        if (dotIndex == -1) {
            throw new DotException("Invalid Dot usage");
        }


        String domain = email.substring(email.lastIndexOf('.') + 1);
```

```
        if (!(domain.equals("in") || domain.equals("com") || domain.equals("net") ||
domain.equals("biz"))) {
            throw new DomainException("Invalid Domain");
        }
    }
}
```

Status : Correct **Marks : 10/10**

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Elsa, a busy professional, is using a scheduling application to plan her meetings efficiently. The application requires users to input meeting durations in minutes, ensuring that the duration is a positive integer and does not exceed 240 minutes (4 hours). Elsa needs a program to assist her in scheduling meetings securely with proper exception handling.

Create a Java class named ElsaMeetingScheduler.Implement a custom exception:InvalidDurationException for invalid meeting duration entries.Implement the main method to interactively take user input for a meeting duration.Implement the validateMeetingDuration method to validate the meeting duration based on the specified rules and throw a custom exception if the validation fails.Print appropriate success or error messages based on the meeting duration.

Implement a custom exception, InvalidDurationException, to handle cases where the entered meeting duration does not meet the specified criteria.

Input Format

The input consists of an integer value 'n', representing the meeting duration.

Output Format

The output is displayed in the following format:

If the entered meeting duration meets the specified criteria, the program outputs

"Meeting scheduled successfully!"

If the entered meeting duration is invalid, the program outputs an error message indicating the issue.

"Error: Invalid meeting duration. Please enter a positive integer not exceeding 240 minutes (4 hours)."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 120
Output: Meeting scheduled successfully!

Answer

```
// You are using Java
import java.util.Scanner;

class InvalidDurationException extends Exception {
    public InvalidDurationException(String message) {
        super(message);
    }
}

class ElsaMeetingScheduler {

    public static void validateMeetingDuration(int duration) throws
```

```
InvalidDurationException {
        if (duration <= 0 || duration > 240) {
            throw new InvalidDurationException(
                "Invalid meeting duration. Please enter a positive integer not exceeding
240 minutes (4 hours)."
            );
        }
    }

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

        try {
            validateMeetingDuration(n);
            System.out.println("Meeting scheduled successfully!");
        } catch (InvalidDurationException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Status : Correct **Marks : 10/10**

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 9_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Bobby is tasked with processing a sequence of numbers from a monitoring system. He needs to extract a strictly increasing subsequence using an ArrayList. The program should dynamically add numbers to the ArrayList only if they are greater than the last number currently stored in the list. Bobby aims to efficiently utilize the dynamic resizing and indexing features of the ArrayList to solve this problem.

Help Bobby implement this solution.

Input Format

The first line of input consists of an integer N, representing the number of elements.

The second line consists of N space-separated integers, representing the elements.

Output Format

The output prints the list of integers in increasing sequence, ignoring out-of-order elements.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 7
3 5 9 1 11 7 13
Output: [3, 5, 9, 11, 13]

Answer

```
// You are using Java
import java.util.*;

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

        int N = sc.nextInt();
        ArrayList<Integer> list = new ArrayList<>();

        for (int i = 0; i < N; i++) {
            int num = sc.nextInt();

            if (list.isEmpty()) {
                list.add(num);
            }

            else if (num > list.get(list.size() - 1)) {
                list.add(num);
            }
        }

        System.out.println(list);
        sc.close();
```

```
    }
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 9_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Vikram loves listening to music and wants to create a simple playlist manager using Java Collections. The playlist supports the following operations:

"ADD <song>" Adds the song to the end of the playlist."REMOVE <song>" Removes the first occurrence of the song from the playlist. If the song is not found, do nothing."SHOW" Displays all songs in the playlist in order. If the playlist is empty, print "EMPTY"."NEXT" Moves to the next song in the playlist and prints its name. If the playlist is empty, print "EMPTY".

The playlist maintains a "current song" position that starts at the first song when it's added. The NEXT command moves to the next song and prints it, wrapping around to the first song after reaching the last song. When removing songs, the current position adjusts accordingly to maintain

proper navigation.

Help Vikram implement this playlist manager.

Input Format

The first line of the input consists of an integer n, the number of operations.

The next n lines, each containing a command:

- "ADD <song>"
- "REMOVE <song>"
- "SHOW"
- "NEXT"

Output Format

For each "SHOW" command, print the songs in order, separated by spaces.

For each "NEXT" command, print the next song in the playlist.

If no song exists, print "EMPTY".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 7
ADD song1
ADD song2
SHOW
NEXT
REMOVE song2
SHOW
NEXT

Output: song1 song2
song2
song1
song1

Answer

```
import java.util.*;

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

        LinkedList<String> playlist = new LinkedList<>();
        int currentIndex = 0;

        for (int i = 0; i < n; i++) {
            String command = sc.nextLine().trim();

            if (command.startsWith("ADD")) {
                String song = command.substring(4);
                playlist.add(song);
            }
            else if (command.startsWith("REMOVE")) {
                String song = command.substring(7);
                int index = playlist.indexOf(song);
                if (index != -1) {
                    playlist.remove(index);

                    if (playlist.isEmpty()) {
                        currentIndex = 0;
                    } else if (index <= currentIndex && currentIndex > 0) {
                        currentIndex--;
                    }
                }
            }
            else if (command.equals("SHOW")) {
                if (playlist.isEmpty()) {
                    System.out.print("EMPTY ");
                } else {
                    for (String s : playlist) {
                        System.out.print(s + " ");
                    }
                }
            }
            else if (command.equals("NEXT")) {
                if (playlist.isEmpty()) {
```

```
                System.out.print("EMPTY ");
            } else {
                currentIndex = (currentIndex + 1) % playlist.size();
                System.out.print(playlist.get(currentIndex) + " ");
            }
        }
    }
    sc.close();
  }
}
```

Status : Correct **Marks : 10/10**

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 10_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : COD

1. Problem Statement

A city traffic management system needs to track vehicles entering a toll booth. Each vehicle is uniquely identified by its registration number. The system should allow adding vehicles to a record, ensuring that no duplicate registration numbers exist. The vehicles should be stored in a HashSet, which does not guarantee any specific order.

Your task is to implement a program using a HashSet that allows adding vehicle details and displaying the records.

Input Format

The first line of input contains an integer N - the number of vehicles.

The next N lines contain details of each vehicle in the format: "RegNumber

OwnerName VehicleType"

1. RegNumber (String) - A unique registration number (Alphanumeric).
2. OwnerName (String) - The name of the vehicle owner.
3. VehicleType (String, Car, Bike, or Truck) - The type of vehicle.

If a vehicle with the same registration number is already present, ignore the duplicate entry.

Output Format

The output prints the unique vehicle records in any order (since HashSet does not maintain order).

Output format: "RegNumber OwnerName VehicleType"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
KA01AB1234 John Car
MH02CD5678 Alice Bike
DL03EF9012 Bob Truck
TN04GH3456 Mike Car
KA01AB1234 John Car
Output: TN04GH3456 Mike Car
KA01AB1234 John Car
MH02CD5678 Alice Bike
DL03EF9012 Bob Truck

Answer

```
// You are using Java
import java.util.*;

class Vehicle {
    private String regNumber;
    private String ownerName;
    private String vehicleType;

    public Vehicle(String regNumber, String ownerName, String vehicleType) {
```

```
        this.regNumber = regNumber;
        this.ownerName = ownerName;
        this.vehicleType = vehicleType;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        Vehicle other = (Vehicle) obj;
        return regNumber.equals(other.regNumber);
    }

    @Override
    public int hashCode() {
        return regNumber.hashCode();
    }

    @Override
    public String toString() {
        return regNumber + " " + ownerName + " " + vehicleType;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();
        sc.nextLine();
        HashSet<Vehicle> vehicles = new HashSet<>();
        for (int i = 0; i < N; i++) {
            String line = sc.nextLine();
            String[] parts = line.split(" ");
            vehicles.add(new Vehicle(parts[0], parts[1], parts[2]));
        }
        for (Vehicle v : vehicles) {
            System.out.println(v);
        }
        sc.close();
    }
}
```

Status : Correct *Marks : 10/10*

Rajalakshmi Engineering College

Name: VISHRUTHI K S
Email: 241801318@rajalakshmi.edu.in
Roll no: 241801318
Phone: 9791109021
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 10_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : COD

1. Problem Statement

John is organizing a fruit festival, and the quantities of various fruits are stored in a HashMap where fruit names are keys and quantities are values.

Help him develop a program to find the total quantity of fruits for the festival by summing up the values in the HashMap.

Input Format

The input consists of fruit quantities in the format 'fruitName:quantity', where fruitName is the name of the fruit(a string), and quantity is a double value representing the quantity.

The input is terminated by entering "done".

Output Format

The output prints a double value, representing the sum of values in the HashMap, rounded off to two decimal places.

If the value is not numeric, print "Invalid input".

If any special characters other than ':' are entered, print "Invalid format".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Banana:15.2
Orange:56.3
Mango:47.3
done

Output: 118.80

Answer

```
// You are using Java
import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        HashMap<String, Double> fruitMap = new HashMap<>();
        double total = 0.0;

        while (true) {
            String input = sc.next();
            if (input.equals("done")) break;
            if (!input.contains(":") || input.indexOf(":") != input.lastIndexOf(":")) {
                System.out.println("Invalid format");
                return;
            }
            String[] parts = input.split(":");
            if (parts.length != 2) {
                System.out.println("Invalid format");
                return;
            }
            String fruit = parts[0];
```

```
            String quantityStr = parts[1];
            try {
                double quantity = Double.parseDouble(quantityStr);
                fruitMap.put(fruit, quantity);
                total += quantity;
            } catch (NumberFormatException e) {
                System.out.println("Invalid input");
                return;
            }
        }

        System.out.printf("%.2f\n", total);
        sc.close();
    }
}
```

Status : Correct **Marks : 10/10**