

RAJALAKSHMI ENGINEERING COLLEGE
RAJALAKSHMI NAGAR, THANDALAM-602 105



**RAJALAKSHMI
ENGINEERING
COLLEGE**

CS23333 OOPS Using Java

Laboratory Record Note Book

Name : M. SABARIPRABU

Year / Branch / Section : II - CSE -

University Register No. : 2116240701446

College Roll No. : 240701446

Semester : III

Academic Year : 2025 - 2026



**RAJALAKSHMI ENGINEERING
COLLEGE**
An Autonomous Institution

BONAFIDE CERTIFICATE

Name: M. Sabariprabu.....

Academic Year: 2025-2026 Semester: III..... Branch: CSE..BE..Computer
Science & Engineering

Register No.

2116240701446 ,

Certified that this is the bonafide record of work done by the above student in
the Object oriented programming Java.....Laboratory
during the academic year 2025- 2026

CS23333

M. Duff 18/11/25
Signature of Faculty in-charge

Submitted for the Practical Examination held on.....

Internal Examiner

External Examiner

INDEX

EX.NO	DATE	NAME OF THE EXPERIMENT	GITHUB QR
1		I/O, Data Types, Operators	
2		Control Structures	
3		Arrays	
4		Strings	
5		Classes & Objects	
6		Inheritance	
7		Interface	
8		Exceptions	
9		Collections	
10		Collections	
11		Project	
12		Lambda	

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 14

Section 1 : MCQ

1. What is the output of the following code?

```
import java.util.*;

class RelationalOperatorExample {
    public static void main(String[] args) {
        int x = 8, y = 4;
        boolean result = (x != y);

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

Answer

true

Status : Correct

Marks : 1/1

2. What will be the output of the following code snippet?

```
import java.util.*;

class OperatorPrecedenceExample {
    public static void main(String[] args) {
        int a = 5, b = 3, c = 2;
        int result = a + b * c;

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

Answer

11

Status : Correct

Marks : 1/1

3. Which of the following data types is used to store floating-point numbers with greater precision?

Answer

double

Status : Correct

Marks : 1/1

4. What is the output of the following code?

```
class TestClass {
    public static void main(String[] args) {
        int a = 5;
        int b = 10;

        int sum = a + b;
        int bitwiseAnd = a & b;
        int bitwiseOr = a | b;
```

```
        System.out.println(sum);
        System.out.println(bitwiseAnd);
        System.out.println(bitwiseOr);
    }
}
```

Answer

15015

Status : Correct

Marks : 1/1

5. Which of the following data types is used to store single characters?

Answer

char

Status : Correct

Marks : 1/1

6. What is the result of the following expression?

```
import java.util.*;

class ComplexExpressionExample {
    public static void main(String[] args) {
        int a = 5, b = 2, c = 3, d = 4;
        int result = a + b * c / d - b;

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

Answer

5

Status : Wrong

Marks : 0/1

7. What will be the output of the following program?

```
class DataTypesMCQ {  
    public static void main(String[] args) {  
        int a = 10;  
        double b = 5;  
        System.out.println(a / b);  
    }  
}
```

Answer

2.0

Status : Correct

Marks : 1/1

8. What is the output of the following program?

```
class Demo {  
    public static void main(String[] args) {  
        String text = "Hello, World!";  
        System.out.println(text);  
    }  
}
```

Answer

Hello, World!

Status : Correct

Marks : 1/1

9. What will be the output of the following code snippet?

```
class DivisionExample {  
    public static void main(String[] args) {  
        double num1 = 10.5;  
        double num2 = 3;  
        int result = (int)(num1 / num2);  
        System.out.println(result);  
    }  
}
```

Answer

3

Status : Correct

Marks : 1/1

10. What is the output of the following program?

```
class Arithmetic {  
    public static void main(String[] args) {  
        char ch = 'A';  
        System.out.println(ch);  
    }  
}
```

Answer

A

Status : Correct

Marks : 1/1

11. What is the output of the following code?

```
class TestClass {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 3;  
        System.out.println(a / b);  
    }  
}
```

Answer

3

Status : Correct

Marks : 1/1

12. Which of the following is not a primitive data type?

Answer

string

Status : Correct

Marks : 1/1

13. What is the output of the following code?

```
class TestClass {  
    public static void main(String[] args) {  
        int x = 5;  
        int X = 10;  
  
        int sum = x + X;  
        int bitwiseResult = x | X;  
  
        System.out.println(sum);  
        System.out.println(bitwiseResult);  
    }  
}
```

Answer

1515

Status : Correct

Marks : 1/1

14. What will be the output of the following code?

```
import java.util.*;  
  
class TernaryOperatorExample {  
    public static void main(String[] args) {  
        int a = 5, b = 10;  
        int result = (a > b) ? a : b;  
        System.out.println(result);  
    }  
}
```

Answer

10

Status : Correct

Marks : 1/1

15. What is the output of the following code?

```
class TestClass {  
    public static void main(String[] args) {  
        int count = 8;  
        count = count ^ 1;  
  
        System.out.println(count);  
    }  
}
```

Answer

9

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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

// You are using Java

```
import java.util.*;
```

```
class Main{
```

```
    public static void main(String arg[]){
```

```
        Scanner sc= new Scanner(System.in);
```

```
        int a=sc.nextInt();
```

```
        int b=sc.nextInt();
```

```
        int a1=Math.abs(100-a);
```

```
        int b1=Math.abs(100-b);
```

```
        if(a1<b1){
```

```
            System.out.println("The integer closer to 100 is" + a + "with a difference  
of"+ a1);
```

```
        }
```

```
        else
```

```
            System.out.println("The integer closer to 100 is" + b + "with a difference  
of"+b1);
```

```
        }
```

```
    }
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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 want help with their doubt. Each hands out an integer and wants to find if one Integer is 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

```
import java.io.*;
import java.util.*;
class Main{
    public static void main(String arg[]){
        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: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem statement

Manoj, a developer at MoneyMatters Inc., is working on improving the company's financial system. He needs to create a program that takes an integer input, converts it into a double, and displays both the original integer and the converted double value.

Input Format

The input consists of a single integer representing a monetary amount.

Output Format

The first line of the output displays the "Original Integer: ", followed by an integer representation of the input value.

The second line displays the "Converted Double: ", followed by a double value representing the input as a decimal value.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 20

Output: Original Integer: 20

Converted Double: 20.0

Answer

```
// You are using Java
import java.util.*;
class main{
    public static void main(String arg[]){
        Scanner sc=new Scanner(System.in);
        int a=sc.nextInt();
        double b=a;
        System.out.println("Original Integer:"+a);
        System.out.println("Converted Double:"+b);
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Vishal and Arun are discussing the properties of numbers. Vishal gives Arun two integers. He asks Arun to check if the sum of these two numbers is a multiple of their product.

Can you assist Arun and determine whether the sum is a multiple of the product?

Input Format

The input consists of two space-separated integers.

Output Format

The output prints:

1. "Sum is Multiple of Product" if the sum of the two numbers is divisible by their product.
2. "Sum is Not Multiple of Product" otherwise.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1 2

Output: Sum is Not Multiple of Product

Answer

```
// You are using Java
import java.util.*;
class Main{
    public static void main(String arg[]){
        Scanner sc=new Scanner(System.in);
        int a=sc.nextInt();
        int b=sc.nextInt();
        int c=a+b;
        int d=a*b;
        if(c==d){
            System.out.println("Sum is Multiple of Product");
        }else{
            System.out.println("Sum is Not Multiple of Product");
        }
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement:

Emily has a beautiful circular garden in her backyard. She's interested in calculating two important measurements for her garden: the circumference and the area. To do this, she needs a program that can take the radius of her circular garden as input and provide the calculated circumference and area as output. The formulas she should use are as follows:

To calculate the circumference (C) of a circle, you can use the formula:

$$C = 2 * \pi * r$$

$$A = \pi * r^2$$

Where:

C represents the circumference.

A represents the area.

π (pi) is approximately 3.14159.

r is the radius of the circle.

Emily is not a programmer, and she needs your help to create a program that will make these calculations for her garden.

Input Format

The first line of input contains a single double-point number radius, representing the radius of the circle.

Output Format

The output should consist of two lines:

The first line should print the circumference of the circle rounded to 2 decimal places, followed by the unit "meters".

The second line should print the area of the circle rounded to 2 decimal places, followed by the unit "square meters".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3.0

Output: Circumference: 18.85 meters

Area: 28.27 square meters

Answer

```
// You are using Java
import java.util.*;
class Main{
    public static void main(String arg[]){
        Scanner sc=new Scanner(System.in);
        double r=sc.nextDouble();
        double c=2*3.14159*r;
```

```
double a=3.14159*(r*r);  
System.out.printf("Circumference: %.2f meters\n",c);  
System.out.printf("Area: %.2f square meters",a);  
}  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q6

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Joey is learning about bitwise operations and is working on a project that involves extracting specific bits from integers. He needs to write a program that takes an integer and the number of bits N as input and outputs the value of the lowest N bits of the integer.

Help Joey in his project to understand and visualize how bitwise operations work in practical scenarios.

Input Format

The first line of input consists of an integer X, representing the given integer.

The second line consists of an integer N, representing the number of bits to extract.

Output Format

The output displays "Result: " followed by an integer representing the value of the lowest N bits of the given integer.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 85

2

Output: Result: 1

Answer

```
// You are using Java
import java.util.*;
class Main{
    public static void main(String arg[]){
        Scanner sc=new Scanner(System.in);
        int a=sc.nextInt();
        int b=sc.nextInt();
        int c=(1<<b)-1;
        int d=a&c;
        System.out.printf("Result: %d",d);
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q7

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement:

Miles is working on a program that involves analyzing two integers. He wants to check if either one of the integers is both:

Less than or equal to zero, and Odd. Can you help him create a program that identifies whether either of the integers meets these conditions?

Input Format

The input consists of two integers on separate lines, denoted as 'input1' and 'input2'.

Output Format

A single line with a boolean result (either 'true' or 'false') indicating whether either 'input1' or 'input2' is both less than or equal to zero and odd.

Refer to the sample output for format specifications

Sample Test Case

Input: -45

10

Output: true

Answer

```
// You are using Java
import java.util.*;
class Main{
    public static void main(String arg[]){
        Scanner sc = new Scanner(System.in);
        int a=sc.nextInt();
        int b=sc.nextInt();
        boolean c=((a<=0) && (a%2!=0)) || ((b<=0) && (b%2!=0));

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

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q8

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In the Kingdom of Finance, the royal treasury is managed by the treasurer, Sir Cedric. Sir Cedric tracks the daily expenses of the kingdom using an expense report that lists three major categories: food, clothing, and utilities. However, the King wants to know if the average daily expense is greater than at least two of these categories to ensure the kingdom is spending wisely.

Your task is to help Sir Cedric determine if the average daily expense is greater than two of the categories. Specifically, you need to calculate the average of the three expenses and check if it is greater than any two categories.

Note: Use the ternary operator

Input Format

Three integers a, b, and c represent the daily expenses for food, clothing, and utilities. Each integer is provided on a single line.

Output Format

The average of the three expenses, rounded to two decimal places.

A message indicating whether the average is greater than at least two of the expense categories.

1. If the average is greater than the two smallest monthly expenses, print "Average is greater than both X and Y," where X and Y are the two smallest expenses.
2. Otherwise, display "Average is not greater than two smallest expenses".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 4

6

10

Output: 6.67

Average is greater than both 4 and 6

Answer

```
// You are using Java
import java.util.*;
class Main{
    public static void main(String arg[]){
        Scanner sc=new Scanner(System.in);
        int a=sc.nextInt();
        int b=sc.nextInt();
        int c=sc.nextInt();
        double d=(double)(a+b+c)/3;
        System.out.printf("%.2f",d);
        if(d>a && b<d){
            System.out.printf("Average is greater than both %d and %d ",a,b);
        }else if(d>a && c<d){
```

```
        System.out.printf("Average is greater than both %d and %d ",a,c);
    }else if(d>b && c<d){
        System.out.printf("Average is greater than both %d and %d ",b,c);
    }else{
        System.out.printf("Average is not greater than two smallest expenses");
    }
}
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q9

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Phill is a quality control manager at a manufacturing plant. He needs to verify if a sensor reading at a midpoint station (S2) falls exactly halfway between the readings of the previous station (S1) and the next station (S3). Help him by developing a program that checks if the second sensor reading is the average (midpoint) of the first and third sensor readings.

Use the relational operator to solve the program.

Input Format

The first line of input consists of an integer S1, representing the sensor reading of the first station.

The second line consists of an integer S2, representing the sensor reading of the midpoint station.

The third line consists of an integer S3, representing the sensor reading of the next station.

Output Format

The first line of output displays a boolean value representing whether the sensor reading at the midpoint station is halfway between the readings of the first and the next stations.

The second line displays one of the following:

1. If the result is true, print "The second integer is halfway between the first and third integers."
2. Otherwise, print "The second integer is not halfway between the first and third integers."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1

7

10

Output: false

The second integer is not halfway between the first and third integers.

Answer

```
// You are using Java
import java.util.*;
class Main{
    public static void main(String arg[]){
        Scanner sc=new Scanner(System.in);
        int a=sc.nextInt();
        int b=sc.nextInt();
        int c=sc.nextInt();

        if((a+c)/2==b){
            System.out.println("true");
            System.out.println("The second integer is halfway between the first and
third integers.");
        }
    }
}
```

```
}else{
    System.out.println("false");
    System.out.println("The second integer is not halfway between the first
and third integers.");
}
}
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 1_Q10

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Aishu is supervising a construction project that needs to be completed with the help of three workers: A, B, and C.

She knows how many days each of them would take to complete the entire project individually:

A can complete it in x days, B in y days, C in z days.

Initially, all three workers (A, B, and C) work together for d_1 days.

After that, C leaves, and only A and B continue for another d_2 days.

Then B also leaves, and A works alone to finish the remaining work.

Your task is to help aishu to implement this functionality using the class `WorkDistribution` and Method `calculateWork(int x, int y, int z, int d1, int d2)`

Calculate the total work completed in the first d_1 days by A, B, and C. Calculate the work completed in the next d_2 days by A and B. Determine the remaining work after these $d_1 + d_2$ days.

Input Format

The first line of input contains five space-separated integers: x y z d_1 d_2

where:

x represents the Days A takes to complete the work alone

y represents the Days B takes to complete the work alone

z represents the Days C takes to complete the work alone

d_1 represents the Days A, B, and C work together

d_2 represents the Days A and B work together (after C leaves)

Output Format

The first line of output prints "Work done in first d_1 days (A+B+C): " followed by a double value rounded to 2 decimal places.

The second line of output prints "Work done in next d_2 days (A+B): " followed by a double value rounded to 2 decimal places.

The third line prints "Remaining work: " followed by a double value rounded to 2 decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 10 20 30 2 2

Output: Work done in first d_1 days (A+B+C): 0.37

Work done in next d_2 days (A+B): 0.30

Remaining work: 0.33

Answer

// You are using Java

```
import java.util.*;
```

```
class Main{
```

```
    public static void main(String arg[]){
```

```
        Scanner sc=new Scanner (System.in);
```

```
        int a =sc.nextInt();
```

```
        int b =sc.nextInt();
```

```
        int c =sc.nextInt();
```

```
        int d1 =sc.nextInt();
```

```
        int d2 =sc.nextInt();
```

```
        double x=1.0/a;
```

```
        double y=1.0/b;
```

```
        double z=1.0/c;
```

```
        double w1=d1*(x+y+z);
```

```
        double w2=d2*(x+y);
```

```
        double r=1.0-(w1+w2);
```

```
        System.out.printf("Work done in first d1 days (A+B+C):%.2f\n",w1);
```

```
        System.out.printf("Work done in next d2 days (A+B):%.2f\n",w2);
```

```
        System.out.printf("Remaining work: %.2f",r);
```

```
    }  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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;
```

```
        System.out.println("Average score:"+ avg);
```

```
        if(avg>=50){
```

```
            System.out.println("The student has passed");
```

```
        }else{
```

```
            System.out.println("The student has failed");
```

```
        }
```

```
    }
```

```
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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 a=sc.nextInt();
```

```
        if((a%5==0)&&(a%7!=0)){
```

```
            System.out.printf("%d is a multiple of 5",a);
```

```
        }else if(a%7==0){
```

```
            System.out.printf("%d is a multiple of 7",a);
```

```
        }else{
```

```
            System.out.printf("%d is neither multiple of 5 nor 7",a);
```

```
        }
```

```
    }
```

```
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. 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 = $\text{weight}/(\text{height}*\text{height})$

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

```
// You are using Java
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 BMI=b/(a*a);
        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: NormalWeight");
        }else if((BMI>=25.0) && (BMI<=29.9)){
            System.out.println("Classification:Overweight");
        }else if(BMI>=30.0){
            System.out.println("Classification: Obese");
        }
    }
}
```

```
}else{  
    System.out.println("nothing to print");  
}  
}  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 9

Section 1 : Coding

1. Problem Statement

Amit wants to evaluate the depreciation of his car over time to understand its current value and categorize it based on that value.

Write a program that helps him determine the current value of his car after a certain number of years of depreciation and classify it into one of three categories:

High: If the current value is greater than 10,000. Medium: If the current value is between 5,000 and 10,000, both inclusive. Low: If the current value is less than 5,000.

The depreciation rate of the car is 15% per year. The program should calculate the current value of the car after applying this depreciation over the given number of years and print the current value along with the category.

Input Format

The first line of input consists of an integer, representing the initial cost of the car.

The second line consists of an integer, representing the number of years the car has been depreciating.

Output Format

The first line of output prints a double value, representing the current value of the car, rounded off to two decimal places "Current Value: <value>".

The second line prints its category "Category: <categories>".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 20000
5

Output: Current Value: 8874.11
Category: Medium

Answer

```
// You are using Java
import java.util.*;
class Main{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        int c=sc.nextInt();
        int y=sc.nextInt();
        double d=(double)15/100;
        double cv=c*Math.pow((1-d),y);
        System.out.printf("Current Value: %.2f\n",cv);
        if(cv>10000){
            System.out.println("Category: High");
        }else if((cv>5000)&&(cv<10000)){
            System.out.println("Category: Medium");
        }else if(cv<5000){
            System.out.println("Category: Low");
        }
    }
}
```

```
}else{  
    System.out.println("nothing");  
}  
}  
}
```

Status : Partially correct

Marks : 9/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. 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.*;
```

```
class Main{
```

```
    public static void main(String args[]){
```

```
        Scanner sc=new Scanner(System.in);
```

```
        int n=sc.nextInt();
```

```
        int count=0,sum=0,a=n;
```

```
        do{
```

```
            count++;
```

```
            sum=sum+a%10;
```

```
            a=a/10;
```

```
        }while(a!=0);
```

```
        if(count==sum){
```

```
            System.out.printf("The number of digits in %d matches the sum of its  
digits.",n);
```

```
        }else{
```

```
            System.out.printf("The number of digits in %d does not match the sum of  
its digits.",n);
```

```
        }
```

```
    }
```

```
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q6

Attempt : 1
Total Mark : 10
Marks Obtained : 10

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

```
int n=sc.nextInt();
for(int i=0;i<n;i++){
    for(int j=0;j<=i;j++){
        System.out.print("* ");
    }
    System.out.printf("\n");
}
for(int i=1;i<n;i++){
    for(int j=n;j>i;j--){
        System.out.print("* ");
    }
    System.out.printf("\n");
}
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q7

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are taking part in a coding challenge where your task is to design a program that conjures a mesmerizing numerical pyramid pattern. The enchanting pattern is fashioned using a for loop and is customized based on user input.

Participants are prompted to unveil the pyramid's magic by specifying its height - essentially dictating the number of rows in this spellbinding creation.

Write a program that employs to weave this captivating numerical pyramid as shown below.

Example

Input:

4

Output:

Input Format

The input consists of a positive integer n representing the number of rows in the pattern.

Output Format

The output prints the required pyramid pattern, as shown in the sample output.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 4

Output: 1

123

12345

1234567

Answer

```
// You are using Java
import java.util.*;
class Main{
    public static void main(String arg[]){
        Scanner sc=new Scanner (System.in);
        int n=sc.nextInt();
        for(int i=1;i<=n;i++){
            for(int j=1;j<i*2;j++){
                System.out.print( j );
            }
            System.out.printf("\n");
        }
    }
}
```

}

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 2_Q8

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

A bank generates secure codes using 3-digit numbers where each digit is unique, and the code must be divisible by 3. You are tasked with generating the first N such codes based on user input, ensuring the digits are unique and the number is divisible by 3.

Note: Use nested for loops to solve.

Input Format

The first line contains an integer N representing the number of valid codes to generate.

Output Format

The output prints N lines, each line contains a valid 3-digit code.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

Output: 102

105

108

120

123

Answer

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int N = scanner.nextInt();
        int count = 0;
        for (int i = 1; i <= 9; i++) {
            for (int j = 0; j <= 9; j++){
                for (int k = 0; k <= 9; k++) {
                    if (i != j && i != k && j != k) {
                        int num = i * 100 + j * 10 + k;
                        if (num % 3 == 0) {
                            System.out.println(num);
                            count++;
                            if (count == N){
                                break;
                            }
                        }
                    }
                }
            }
            if (count == N) break;
        }
        if (count == N) break;
    }
}
```

}

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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

```
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 res=arr[1]+arr[n-3];
        System.out.println(res);

    }

}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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 mainSum=0,secSum=0;
for(int i=0;i<n;i++)
{

    mainSum+=arr[i][i];
    secSum+=arr[i][n-1-i];

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

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 3_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are developing a warehouse management system for a shipping company. The system uses an integer array to represent the weights of packages in a specific order. To verify that the weight capacity is not exceeded, the program needs to calculate the sum of the weights of the first and last packages in the list.

Task:

Write a code to calculate the sum of the weights of the first and last packages in the list. The program should take an integer array as input and return the total weight of the first and last packages.

Input Format

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

The second line of the input is N space-separated integer values.

Output Format

The output is displayed in the following format:

"Sum of the first and last elements: <<Sum>>"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

10 20 30 40 50

Output: Sum of the first and last elements: 60

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();

int sum=arr[0]+arr[n-1];

System.out.println("Sum of the first and last elements: "+sum);

}

}

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 3_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Sesha is developing a weather monitoring system for a region with multiple weather stations. Each weather station collects temperature data hourly and stores it in a 2D array.

Write a program that can add the temperature data from two different weather stations to create a combined temperature record for the region.

Input Format

The first line of input consists of two space-separated integers N and M, representing the number of rows and columns of the matrices, respectively.

The next N lines consist of M space-separated integers, representing the values of the first matrix.

The following N lines consist of M space-separated integers, representing the values of the second matrix.

Output Format

The output prints the addition of the two matrices in N rows and M columns, representing the combined temperature record.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3 3

1 2 3

4 5 6

7 8 9

1 1 1

2 2 2

3 3 3

Output: 2 3 4

6 7 8

10 11 12

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(),m=sc.nextInt();
    int[][] a=new int[n][m];
    int[][] b=new int[n][m];
    for(int i=0;i<n;i++)
```

```
        for(int j=0;j<m;j++)
            a[i][j]=sc.nextInt();
        for(int i=0;i<n;i++)
            for(int j=0;j<m;j++)
                b[i][j]=sc.nextInt();
        for(int i=0;i<n;i++)

    {

        for(int j=0;j<m;j++)
            System.out.print((a[i][j]+b[i][j])+" ");
        System.out.println();
    }

}

}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 3_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Sharon is creating a program that finds the first repeated element in an integer array. The program should efficiently identify the first element that appears more than once in the given array. If no such element is found, it should appropriately display a message.

Help Sharon to complete the program.

Input Format

The first line of input consists of an integer n , representing the number of elements in the array.

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

Output Format

If a repeated element is found, print the first element that appears more than once.

If no repeated element is found, print "No repeated element found in the array".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 8

12 21 13 14 21 36 47 21

Output: 21

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();

HashSet<Integer> set=new HashSet<>();

boolean found=false;

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

if(set.contains(arr[i])){

System.out.println(arr[i]);

found=true;

break;

}

set.add(arr[i]);

}

if(!found) System.out.println("No repeated element found in the array");

}

}

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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

```
        for(int i=0;i<n;i++){
```

```
            String passage=sc.nextLine();
```

```
            int commacount=0,periodcount=0,questioncount=0;
```

```
            for(int j=0;j<passage.length();j++){
```

```
                char c=passage.charAt(j);
```

```
                if(c==','){
```

```
                    commacount++;
```

```
                }else if(c=='.'){
```

```
                    periodcount++;
```

```
                }else if(c=='?'){
```

```
                    questioncount++;
```

```
                }
```

```
            }
```

```
            System.out.printf("%d %d %d\n",commacount,periodcount,questioncount);
```

```
        }
```

```
    }
```

```
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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();
        sc.nextLine();
        String[] keywords=sc.nextLine().split(" ");
        Arrays.sort(keywords);
        for(int i=0;i<n;i++){
            System.out.println(keywords[i]);
            if(i!=n-1){
                System.out.print(" ");
            }
        }
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 4_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Bechan Chacha is seeking help to filter out valid mobile numbers from a list provided by his crush. He can only pick his crush's number if the list contains valid mobile numbers.

A mobile number is considered valid if:

It has exactly 10 digits. It consists only of numeric values (0–9). It does not begin with zero.

Your task is to determine whether each mobile number in the list is valid or not.

Input Format

The first line contains an integer T, representing the number of mobile numbers

to check.

The next T lines each contain a string S, representing a mobile number.

Output Format

For each mobile number S, the output print "YES" if it is valid.

Otherwise, print "NO".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
9876543210
Output: YES

Answer

// You are using Java

import java.util.*;

class main{

 public static void main(String args[]){

 Scanner sc=new Scanner(System.in);

 int T=sc.nextInt();

 sc.nextLine();

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

 String number=sc.nextLine().trim();

 boolean isValid=true;

 if(number.length()!=10){

 isValid=false;

 }

 for(int j=0;j<number.length();j++){

 if(!Character.isDigit(number.charAt(j))){

 isValid=false;

 }

 }

 if(number.length()>0 && number.charAt(0)=='0'){

 isValid=false;

 }

 if(isValid){

```
        System.out.printf("Yes");
    }else{
        System.out.printf("No");
    }
}
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 4_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Arjun is learning how to filter words from a sentence based on grammar rules. He wants to identify the valid words in a sentence.

A word is considered valid if it satisfies all these conditions:

The word contains only alphabets (a-z, A-Z). The word length is at least 2 characters. The word should not contain digits or special characters.

Your task is to read a sentence and print all the valid words in it.

Input Format

The input contains a single line containing a sentence S.

Output Format

The output prints all the valid words separated by spaces.

If no valid word exists, print "No valid words."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Hello world1 123 ab" @#\$ Hi

Output: Hello Hi

Answer

// You are using Java

import java.util.Scanner;

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

        String[] words = sentence.split(" ");
        StringBuilder validWords = new StringBuilder();
        boolean foundValidWord = false;

        for (String word : words){
            if (isValid(word)){
                if (foundValidWord){
                    validWords.append(" ");
                }
                validWords.append(word);
                foundValidWord = true;
            }
        }
        if (foundValidWord){
            System.out.println(validWords.toString());
        } else {
            System.out.println("No valid words.");
        }
    }
}
```

```
private static boolean isValid(String word) {  
    if (word.length() < 2){  
        return false;  
    }  
    for (char c : word.toCharArray()){  
        if (!Character.isLetter(c)){  
            return false;  
        }  
    }  
  
    return true;  
}  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 4_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In a secure banking system, customers are required to create PIN codes for accessing their accounts. The bank wants to validate these PIN codes before accepting them.

A PIN code is considered valid if:

It consists of exactly 4 digits. All characters must be numeric (0–9). It cannot contain all identical digits (e.g., 1111 is invalid).

Your task is to determine whether each PIN code in the list is valid or not.

Input Format

The first line of input contains an integer T, representing the number of PIN codes to check.

The next T lines each contain a string S, representing a PIN code.

Output Format

For each PIN code S, the output print "YES" if it is valid.

Otherwise, the output print "NO".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1

1234

Output: YES

Answer

```
import java.util.Scanner;

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

        for (int i = 0; i < T; i++){
            String pin = scanner.nextLine();
            if (isValidPin(pin)){
                System.out.println("YES");
            }else{
                System.out.println("NO");
            }
        }
        scanner.close();
    }
    private static boolean isValidPin(String pin){
        if (pin.length() != 4){
            return false;
        }
        for (char c : pin.toCharArray()){
```

```
        if (!Character.isDigit(c)){  
            return false;  
        }  
    }  
    char firstDigit = pin.charAt(0);  
    for (int i = 1; i < pin.length(); i++){  
        if (pin.charAt(i) != firstDigit){  
            return true;  
        }  
    }  
  
    return false;  
}  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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 Account {
```

```
    int number;
```

```
    String name;
```

```
    double balance;
```

```
    Account(int number, String name, double balance){
```

```
        this.number = number;
```

```
        this.name = name;
```

```
        this.balance = balance;
```

```
    }
```

```
    void deposit(double amt){
```

```
        balance += amt;
```

```
    }
```

```
    void withdraw(double amt){
```

```
        if (amt <= balance) balance -= amt;
```

```
    }
```

```
}
```

```
public class Main {
```

```
    public static void main(String[] args){
```

```
        Scanner sc = new Scanner(System.in);
```

```
        int n = sc.nextInt();
```

```
        sc.nextLine();
```

```
        for (int i = 0; i < n; i++) {
```

```
            int num = Integer.parseInt(sc.nextLine());
```

```
            String cname = sc.nextLine();
```

```
            double bal = Double.parseDouble(sc.nextLine());
```

```
            double dep = Double.parseDouble(sc.nextLine());
```

```
            double wit = Double.parseDouble(sc.nextLine());
```

```
            Account a = new Account(num, cname, bal);
```

```
            a.deposit(dep);
```

```
            a.withdraw(wit);
```

```
System.out.println("Account Number: " + a.number);  
System.out.println("Customer Name: " + a.name);  
System.out.println("Final Balance: " + String.format("%.1f", a.balance));
```

```
}  
}
```

```
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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 unit For the next 100 units (101–200) 7 units charge per unit For units above 200 10 units charge per unit If 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

// You are using Java

```
import java.util.*;
class Customer{
    int id;
    String name;
    double units;

    Customer(int id, String name, double units){
        this.id = id;
        this.name = name;
        this.units = units;
    }
    double calculateBill(){
        double bill = 0;
        if (units <= 100) {
            bill = units * 5;
        } else if (units <= 200){
            bill = 100 * 5 + (units - 100) * 7;
        } else {
            bill = 100 * 5 + 100 * 7 + (units - 200) * 10;
        }
        if (bill > 2000){
            bill = bill - bill * 0.05;
        }
        return bill;
    }
}
```

```
public class Main {
    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 cid = Integer.parseInt(sc.nextLine());
            String cname = sc.nextLine();
            double units = Double.parseDouble(sc.nextLine());

            Customer c = new Customer(cid, cname, units);
            double finalBill = c.calculateBill();
        }
    }
}
```

```
System.out.println("Customer ID: " + c.id);  
System.out.println("Customer Name: " + c.name);  
System.out.println("Final Bill: " + String.format("%.1f", finalBill));
```

```
}  
    sc.close();
```

```
}
```

```
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are working as a developer for CityCab, a taxi service company that wants to build a ride fare management system.

Each customer booking has:

A Booking ID (integer) A Customer Name (string) A Distance Travelled in km (double)

The fare calculation rules are:

Base Fare = 50 units (flat charge for every ride). Per km charge = 10 units/km. If the distance is greater than 20 km, a 10% discount is applied on the total fare.

You are required to implement this system using:

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

Finally, display each booking's details and final fare.

Input Format

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

For each booking:

- The next line contains the booking ID (integer).
- The following line contains the customer's name (string).
- The next line contains the distance travelled (double).

Output Format

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

1. Booking ID: <booking_id>
2. Customer Name: <customer_name>
3. Final Fare: <final_fare> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1

1234

Rahul Sharma

15

Output: Booking ID: 1234

Customer Name: Rahul Sharma

Final Fare: 200.0

Answer

```
import java.util.*;
class Booking {
    int bookingId;
```

```
String customerName;  
double distance;
```

```
Booking(int bookingId, String customerName, double distance)
```

```
{
```

```
    this.bookingId = bookingId;  
    this.customerName = customerName;  
    this.distance = distance;
```

```
}
```

```
    int getBookingId()
```

```
{
```

```
    return bookingId;
```

```
}
```

```
    String getCustomerName()
```

```
{
```

```
    return customerName;
```

```
}
```

```
    double getDistance()
```

```
{
```

```
    return distance;
```

```
}
```

```
    public void setBookingId(int id)
```

```
{
```

```
        bookingId = id;
```

```
}  
    public void setCustomerName(String name)
```

```
{
```

```
    customerName = name;
```

```
}
```

```
    public void setDistance(double dist)
```

```
{
```

```
    distance = dist;
```

```
}
```

```
    double calculateFare()
```

```
{
```

```
        double fare = 50 + distance * 10;
```

```
        if (distance > 20) fare *= 0.9;
```

```
        return fare;
```

```
    }
```

```
}
```

```
class Main
```

```
{
```

```
    public static void main(String[] args)
```

```
{
```

```
        Scanner sc = new Scanner(System.in);
```

```
int n = Integer.parseInt(sc.nextLine().trim());
for (int i = 0; i < n; i++)
{

    int id = Integer.parseInt(sc.nextLine().trim());
    String name = sc.nextLine().trim();
    double dist = Double.parseDouble(sc.nextLine().trim());
    Booking b = new Booking(id, name, dist);
    System.out.println("Booking ID: " + b.getBookingId());
    System.out.println("Customer Name: " + b.getCustomerName());
    System.out.println("Final Fare: " + String.format("%.1f", b.calculateFare()));

}

    sc.close();

}

}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 5_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Ram is working as a developer for BrightEdu Coaching Center, which wants to build a student fee management system.

Each student's enrollment has:

An Enrollment ID (integer) A Student Name (string) The Number of Subjects (integer)

The fee calculation rules are:

Registration Fee = 1000 units (flat for every student). Per Subject Fee = 800 units. If the student enrolls in more than 5 subjects, a 20% scholarship (discount) is applied on the total fee.

Ram has been asked to implement this system using:

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

Finally, display each student's details and final fee.

Input Format

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

For each student:

- The next line contains the Enrollment ID (integer).
- The following line contains the student's name (string).
- The next line contains the Number of subjects (integer).

Output Format

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

- Enrollment ID: <enrollment_id>
- Student Name: <student_name>
- Final Fee: <final_fee> (rounded to one decimal place)

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1

1234

Ravi Kumar

3

Output: Enrollment ID: 1234

Student Name: Ravi Kumar

Final Fee: 3400.0

Answer

```
// You are using Java
import java.util.*;
class Student{
    int enrollmentId;
```

```

String name;
int subjects;

Student(int enrollmentId, String name, int subjects){
    this.enrollmentId = enrollmentId;
    this.name = name;
    this.subjects = subjects;
}
double calculateFee(){
    double fee = 1000 + subjects * 800;
    if (subjects > 5){
        fee = fee - fee * 0.20;
    }
    return fee;
}

}public class Main {
    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 sname = sc.nextLine();
            int subs = Integer.parseInt(sc.nextLine());

            Student st = new Student(id, sname, subs);
            double finalFee = st.calculateFee();

            System.out.println("Enrollment ID: " + st.enrollmentId);
            System.out.println("Student Name: " + st.name);
            System.out.println("Final Fee: " + String.format("%.1f", finalFee));
        }
        sc.close();
    }
}

```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 6_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 10

Section 1 : MCQ

1. Which of the following is the correct way for class B to inherit from class A?

Answer

class B extends A {}

Status : Correct

Marks : 1/1

2. What will be the output of the following Java program?

```
class Test {  
    void show(int a) {  
        System.out.println("Integer method");  
    }  
    void show(String s) {
```

```
        System.out.println("String method");
    }
    public static void main(String[] args) {
        Test obj = new Test();
        obj.show(null);
    }
}
```

Answer

Compilation error due to ambiguous method call

Status : Wrong

Marks : 0/1

3. What will be the output of the following Java program?

```
class Vehicle {
    void startEngine() {
        System.out.println("Vehicle engine started");
    }
}
```

```
class Car extends Vehicle {
    void startEngine() {
        System.out.println("Car engine started");
    }
}
```

```
class Main {
    public static void main(String[] args) {
        Vehicle myVehicle = new Car();
        myVehicle.startEngine();
    }
}
```

Answer

Car engine started

Status : Correct

Marks : 1/1

4. What will be the output of the following program?

```
class A {  
    int x = 10;  
}  
  
class B extends A {  
    int x = 20;  
}  
  
class C extends B {  
    int x = 30;  
  
    void display() {  
        System.out.println(x);  
        System.out.println(super.x);  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        C obj = new C();  
        obj.display();  
    }  
}
```

Answer

3020

Status : Correct

Marks : 1/1

5. What will be the output of the following Java program?

```
class Vehicle {  
    void start() {  
        System.out.println("Vehicle starts");  
    }  
}  
  
class Car extends Vehicle {
```

```
void start() {  
    System.out.println("Car starts");  
}  
}  
class ElectricCar extends Car {  
    void start() {  
        System.out.println("Electric Car starts silently");  
    }  
}  
class Test {  
    public static void main(String[] args) {  
        Vehicle v = new ElectricCar();  
        v.start();  
    }  
}
```

Answer

Electric Car starts silently

Status : Correct

Marks : 1/1

6. What will be the output of the following code?

```
class A {  
    int sum(int x) {  
        return x + 2;  
    }  
}  
  
class B extends A {  
    int sum(int x) {  
        return super.sum(x) * 2;  
    }  
}  
  
class C extends B {  
    int sum(int x) {  
        return super.sum(x) - 3;  
    }  
}
```

```
}  
}  
  
class Test {  
    public static void main(String[] args) {  
        C obj = new C();  
        System.out.println(obj.sum(4));  
    }  
}
```

Answer

11

Status : Wrong

Marks : 0/1

7. Select the correct keyword for implementing inheritance through the class.

Answer

extends

Status : Correct

Marks : 1/1

8. What will be the output of the following program?

```
class Vehicle {  
    String type = "Vehicle";  
}
```

```
class Car extends Vehicle {  
    String type = "Car";  
}
```

```
class Test {  
    public static void main(String[] args) {  
        Car c = new Car();  
        System.out.println(c.type);  
    }  
}
```

}

Answer

Car

Status : Correct

Marks : 1/1

9. What will be the output of the following code?

```
class A {  
    void display() {  
        System.out.println("Display A");  
    }  
}  
  
class B extends A {  
    void display() {  
        System.out.println("Display B");  
    }  
}  
  
class C extends B {  
    void display() {  
        super.display();  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        C obj = new C();  
        obj.display();  
    }  
}
```

Answer

Display BDisplay A

Status : Wrong

Marks : 0/1

10. What will be the output of the following program?

```
class A {
    public int i;
    private int j;
}
class B extends A {
    void display() {
        super.j = super.i + 1;
        System.out.println(super.i + " " + super.j);
    }
}
class inheritance {
    public static void main(String args[]) {
        B obj = new B();
        obj.i=1;
        obj.j=2;
        obj.display();
    }
}
```

Answer

2 3

Status : Wrong

Marks : 0/1

11. Which of the following is true about method overriding in Java?

Answer

The method must have the same name, same parameters, and must be in different classes with an inheritance relationship

Status : Correct

Marks : 1/1

12. What will be the output of the following Java program?

```
class Test {
    void display(int a, int b) {
```

```

        System.out.println("Method 1");
    }
    void display(double a, double b) {
        System.out.println("Method 2");
    }
    public static void main(String[] args) {
        Test obj = new Test();
        obj.display(10, 10.0);
    }
}

```

Answer

Method 2

Status : Correct

Marks : 1/1

13. What will be the output of the following Java program?

```

class A {
    int value = 10;
    void display() {
        System.out.println("A's display: " + value);
    }
}
class B extends A {
    int value = 20;
    void display() {
        System.out.println("B's display: " + value);
    }
}
class Test {
    public static void main(String[] args) {
        A obj = new B();
        obj.display();
        System.out.println("Value: " + obj.value);
    }
}

```

Answer

B's display: 20 Value: 20

Status : Wrong

Marks : 0/1

14. What will be the output of the following Java program?

```
class A {  
    void display() {  
        System.out.println("Class A");  
    }  
}  
  
class B extends A {  
    void show() {  
        System.out.println("Class B");  
    }  
}  
  
class C extends B {  
    void print() {  
        System.out.println("Class C");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        C obj = new C();  
        obj.display();  
        obj.show();  
        obj.print();  
    }  
}
```

Answer

Class AClass BClass C

Status : Correct

Marks : 1/1

15. What will be the output of the following Java program?

```
class Parent {  
    void show() {  
        System.out.println("Parent class");  
    }  
}  
class Child extends Parent {  
    void show() {  
        System.out.println("Child class");  
    }  
}  
class Test {  
    public static void main(String[] args) {  
        Parent obj = new Child();  
        obj.show();  
    }  
}
```

Answer

Child class

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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;

class PremiumSubscription{
    double baseMonthlyCost;
    double serviceTax;
    double extraFeatureCost;
    PremiumSubscription(double baseMonthlyCost, double serviceTax, double
extraFeatureCost) {
        this.baseMonthlyCost = baseMonthlyCost;
        this.serviceTax = serviceTax;
        this.extraFeatureCost = extraFeatureCost;
    }
    double calculateMonthlyCost(){
        return baseMonthlyCost + 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: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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;
```

```
class Product{
```

```
    public double price;
```

```
    public Product(double price)
```

```
{
```

```
        this.price = price;
```

```
}
```

```
}
```

```
class DiscountedProduct extends Product
```

```
{
```

```
    private double discountRate;
```

```
    public DiscountedProduct(double price, double discountRate)
```

```
{
```

```
        super(price);
```

```
        this.discountRate = discountRate;
```

```
    }
```

```
    public double calculateSellingPrice()
```

```
{
```

```
        if (discountRate > 1)
```

```
{
```

```
            return -1;
```

```
}
```

```
        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: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Preethi is working on a project to automate sales tax calculations for items in a store. She wants to create a program that takes the price of an item and the sales tax rate as input and calculates the final price of the item after applying the sales tax.

Write a program using the class SalesTaxCalculator, which contains an overloaded method named calculateFinalPrice to handle both integer and double inputs. The program should also include a Main class that takes user input, calls the appropriate method from SalesTaxCalculator, and prints the final price of the item.

Formula Used: Final price = price + ((price * sales tax rate) / 100)

Input Format

The first line of input consists of an integer price (the price of the item for integer inputs).

The second line of input consists of an integer taxRate (the sales tax rate for integer inputs).

The third line of input consists of a double price (the price of the item for double inputs).

The fourth line of input consists of a double taxRate (the sales tax rate for double inputs).

Output Format

The first line of output prints an integer, representing the final price of the item after applying the sales tax for integer inputs (a and b).

The second line prints a double value, representing the final price of the item after applying the sales tax for double-value inputs (m and n), rounded to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 100

10

100.0

5.0

Output: 110

105.00

Answer

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

```
// You are using Java
```

```
class SalesTaxCalculator{
```

```

public static int calculateFinalPrice(int price, int taxRate)
{

    return price + (price * taxRate / 100);

}

public static double calculateFinalPrice(double price, double taxRate)
{

    return price + (price * taxRate / 100);

}

}

class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int intPrice = scanner.nextInt();
        int intTaxRate = scanner.nextInt();
        double doublePrice = scanner.nextDouble();
        double doubleTaxRate = scanner.nextDouble();

        int finalPriceInt = SalesTaxCalculator.calculateFinalPrice(intPrice,
intTaxRate);
        double finalPriceDouble =
SalesTaxCalculator.calculateFinalPrice(doublePrice, doubleTaxRate);

        System.out.println(finalPriceInt);
        System.out.format("%.2f", finalPriceDouble);
    }
}

```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q4

Attempt : 2
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Mr.Kapoor wants to create a program to calculate the volume of a Cuboid and a Cube using method overriding.

Implements a base class Cuboid with attributes for length, width, and height. Include a method calculateVolume() that computes the volume of the cuboid.

Extends the base class with a subclass Cube representing a cube, where all sides are equal. Override the calculateVolume() method in the Cube class to compute the volume of the cube.

The program should take user input for the dimensions of the cuboid and the side length of the cube and display the calculated volumes with two decimal places.

Input Format

The first line of input consists of 3 space-separated double values, representing the cuboid length, width, and height, respectively.

The second line consists of a double value, representing the side length of the cube.

Output Format

The first line of output prints the volume of the cuboid, rounded off to two decimal places.

The second line prints the volume of the cube, rounded off to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 60.0 60.0 60.0
50.0

Output: Volume of Cuboid: 216000.00
Volume of Cube: 125000.00

Answer

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

```
class Cuboid
```

```
{
```

```
    protected double length;  
    protected double width;  
    protected double height;
```

```
    public Cuboid(double length, double width, double height)
```

```
{
```

```
this.length = length;  
this.width = width;  
this.height = height;
```

```
}
```

```
public double calculateVolume()
```

```
{
```

```
    return length * width * height;
```

```
}
```

```
}
```

```
// Subclass
```

```
class Cube extends Cuboid
```

```
{
```

```
    public Cube(double side)
```

```
    {
```

```
        super(side, side, side);
```

```
    }
```

```
    public double calculateVolume()
```

```
    {
```

```

        return length * length * length;
    }

}

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

        double cuboidLength = scanner.nextDouble();
        double cuboidWidth = scanner.nextDouble();
        double cuboidHeight = scanner.nextDouble();

        // Regular object instantiation for Cuboid
        Cuboid cuboid = new Cuboid(cuboidLength, cuboidWidth, cuboidHeight);
        System.out.printf("Volume of Cuboid: %.2f\n", cuboid.calculateVolume());

        double cubeSide = scanner.nextDouble();

        // Upcasting - Using superclass reference for subclass object (DMD)
        Cuboid cube = new Cube(cubeSide); // Upcasting
        System.out.printf("Volume of Cube: %.2f", cube.calculateVolume()); // Calls
        Cube's method dynamically

        scanner.close();
    }
}

```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem statement:

Tim was tasked with developing a grocery shopping app. You have a class hierarchy that includes Item, Produce, and OrganicProduce. Your goal is to calculate the total cost of a shopping list, which may contain a mix of regular produce and organic produce items. Additionally, you need to apply discounts to organic items. Apply a 10% discount on organic produce items

Class Hierarchy:

Item: Base class for all items.

Produce: Subclass of Item for regular produce items.

OrganicProduce: Subclass of Produce for organic produce items.

Input Format

The first line of input consists of an integer, 'n'.

For each 'n' item, the user will provide:

- A string 'type' representing the item type ('Regular' or 'Organic').
- A string 'name' represents the item name.
- A double 'price' represents the item price.

Output Format

The output will display the total cost of the shopping list, including discounts on organic items.

Refer to the sample output for format specifications.

Sample Test Case

Input: 1

Regular Banana 1.99

Output: 1.99

Answer

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

```
    protected String name;  
    protected double price;
```

```
    public Item(String name, double price)
```

```
{
```

```
    this.name = name;
```

```
this.price = price;
```

```
}
```

```
public double calculateCost()
```

```
{
```

```
    return price;
```

```
}
```

```
}
```

```
class Produce extends Item
```

```
{
```

```
    public Produce(String name, double price)
```

```
{
```

```
        super(name, price);
```

```
}
```

```
    @Override
```

```
    public double calculateCost()
```

```
{
```

```
    return price;
```

```
}
```

```
}
```

```
class OrganicProduce extends Produce
```

```
{
```

```
    public OrganicProduce(String name, double price)
```

```
{
```

```
        super(name, price);
```

```
}
```

```
    @Override
```

```
    public double calculateCost()
```

```
{
```

```
        return price * 0.9;
```

```
}
```

```
}
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        int n = sc.nextInt();
```

```
        sc.nextLine(); // Consume newline
```

```
        double totalCost = 0.0;
```

```
        for (int i = 0; i < n; i++) {
```

```
            String type = sc.next();
```

```
            String name = sc.next();
```

```
double price = sc.nextDouble();

    if (type.equals("Regular")) {
        Item item = new Produce(name, price);
        totalCost += item.calculateCost();
    } else if (type.equals("Organic")) {
        Item item = new OrganicProduce(name, price);
        totalCost += item.calculateCost();
    }
}

System.out.printf("%.2f%n", totalCost);
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 7_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 10

Section 1 : MCQ

1. What is the output of the following code?

```
interface A {  
    default void show() {  
        System.out.println("A's Default Method");  
    }  
}
```

```
interface B {  
    default void show() {  
        System.out.println("B's Default Method");  
    }  
}
```

```
class C implements A, B {  
    public void show() {
```

```
        A.super.show();
    }
}

public class Main {
    public static void main(String[] args) {
        C obj = new C();
        obj.show();
    }
}
```

Answer

Compilation Error

Status : Wrong

Marks : 0/1

2. What is the output of the following code?

```
interface A {
    default void show() {
        System.out.println("A's Default Method");
    }
}
```

```
class B {
    public void show() {
        System.out.println("B's Method");
    }
}
```

```
class C extends B implements A {
}
```

```
public class Main {
    public static void main(String[] args) {
        C obj = new C();
        obj.show();
    }
}
```

Answer

B's Method

Status : Correct

Marks : 1/1

3. Consider a class implementing an interface and extending a class, both having a method with the same name. Which method gets called?

Answer

The method from the superclass

Status : Correct

Marks : 1/1

4. Which of the following statements is true regarding default methods in Java interfaces?

Answer

A default method can be overridden in a class implementing the interface.

Status : Correct

Marks : 1/1

5. What is the output of the following code?

```
interface A {  
    static void display() {  
        System.out.println("Static method in A");  
    }  
}
```

```
class B implements A {  
    static void display() {  
        System.out.println("Static method in B");  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {
```

```
        B.display();  
    }  
}
```

Answer

Compilation Error

Status : Wrong

Marks : 0/1

6. What happens when an implementing class does not override a default method from an interface?

Answer

The default method's implementation from the interface will be used.

Status : Correct

Marks : 1/1

7. How can a class explicitly call a default method from an interface if there is a naming conflict?

Answer

Using InterfaceName.super.methodName();

Status : Correct

Marks : 1/1

8. What is the primary purpose of static methods in Java interfaces?

Answer

They allow an interface to provide helper methods without requiring an implementing class.

Status : Correct

Marks : 1/1

9. Which of the following statements about Java interfaces is true?

Answer

An interface cannot contain default or static methods.

Status : Wrong

Marks : 0/1

10. How do you call a static method from an interface MyInterface?

Answer

MyInterface.staticMethod();

Status : Correct

Marks : 1/1

11. Can a Java interface contain both default and static methods?

Answer

Yes, an interface can have both default and static methods.

Status : Correct

Marks : 1/1

12. What is the output of the following code?

```
interface X {  
    default void show() {  
        System.out.println("X's Default Method");  
    }  
}
```

```
interface Y {  
    default void show() {  
        System.out.println("Y's Default Method");  
    }  
}
```

```
class Z implements X, Y {  
    public void show() {  
        System.out.println("Z's Method");  
    }  
}
```

```
public class Main {
```

```
public static void main(String[] args) {  
    Z obj = new Z();  
    obj.show();  
}  
}
```

Answer

Compilation Error

Status : Wrong

Marks : 0/1

13. Which of the following is the correct way to declare an interface in Java?

Answer

```
interface Vehicle { void start();}
```

Status : Correct

Marks : 1/1

14. What is the output of the following code?

```
interface MathOperations {  
    static int square(int x) {  
        return x * x;  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        System.out.println(MathOperations.square(5));  
    }  
}
```

Answer

Compilation error

Status : Wrong

Marks : 0/1

15. If a class implements two interfaces that have the same default method, what must the class do?

Answer

The class must override the method to resolve ambiguity.

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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 : 10

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 sc);  
    void calculateAndDisplayCost();
```

```
}
```

```
class EnergyConsumptionTracker implements CostCalculator
```

```
{
```

```
    private double ratePerUnit;  
    private int numDays;  
    private double[] energyConsumed;
```

```
    public EnergyConsumptionTracker(double ratePerUnit, int numDays)
```

```
{
```

```
        this.ratePerUnit = ratePerUnit;  
        this.numDays = numDays;  
        this.energyConsumed = new double[numDays];
```

```
}
```

```
    public void getEnergyDetails(Scanner sc)
```

```
{
```

```
        for (int i = 0; i < numDays; i++)
```

```
{
```

```
energyConsumed[i] = sc.nextDouble();
```

```
}
```

```
}
```

```
public void calculateAndDisplayCost()
```

```
{
```

```
double totalCost = 0;
```

```
System.out.println("Day-wise Energy Cost:");
```

```
for (int i = 0; i < numDays; i++)
```

```
{
```

```
double dailyCost = energyConsumed[i] * ratePerUnit;
```

```
totalCost += dailyCost;
```

```
System.out.printf("Day %d: Rs. %.2f%n", (i + 1), dailyCost);
```

```
}
```

```
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 : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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: $BMI = \text{weight} / (\text{height} * \text{height})$

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;
```

```
// You are using Java
```

```
interface HealthCalculator
```

```
{
```

```
    double calculateBMI(double weight, double height);
```

```
}
```

```
class BMICalculator implements HealthCalculator
```

```
{
```

```
    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: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

A financial analyst, Alex, needs a program to calculate simple interest for various financial transactions. He requires a straightforward tool that takes in the principal amount, interest rate, and time in years and computes the interest.

The formula to be used is: $\text{Interest} = \text{Principal} \times \text{Rate} \times \text{Time} / 100$

Implement this functionality using the InterestCalculator interface and the SimpleInterestCalculator class.

Input Format

The first line of input consists of the principal amount P as a double value.

The second line of input consists of the annual interest rate r as a double value.

The third line of input consists of the number of years t as a positive integer, which is an integer value.

Output Format

The output displays the calculated simple interest in the following format: "Simple Interest: [interest_value]", Here, [interest_value] should be replaced with the actual interest value calculated by the program.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1000.00

5.00

2

Output: Simple Interest: 100.0

Answer

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

```
interface InterestCalculator
```

```
{
```

```
    double simpleInterest(double principal, double rate, int time);
```

```
}
```

```
class SimpleInterestCalculator implements InterestCalculator
```

```
{
```

```
    public double simpleInterest(double principal, double rate, int time)
```

```
{  
  
    return (principal * rate * time) / 100;  
  
}  
  
}
```

```
class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        double principal = scanner.nextDouble();  
  
        double rate = scanner.nextDouble();  
  
        int time = scanner.nextInt();  
  
        InterestCalculator calculator = new SimpleInterestCalculator();  
  
        double interest = calculator.simpleInterest(principal, rate, time);  
  
        System.out.println("Simple Interest: " + interest);  
    }  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Maria, a software developer, is working on an inventory management system project using Java that utilizes an inventory interface to manage a store's products.

The interface should define two methods: `addProduct`, which adds a product by accepting its name, price, and quantity, and `calculateTotalValue`, which computes the total value of all products in the inventory. Implement the interface in a class called `SimpleInventory`, which internally manages a list of `Product` objects.

Each `Product` object should encapsulate the product's name, price, and quantity and include a method to calculate its value as $\text{price} \times \text{quantity}$. The system should allow users to dynamically add products to the inventory and calculate the total value of all products stored.

Help Maria achieve the task.

Input Format

The first line of input consists of an integer to choose one of the following options:

- 1 - to add a product to the inventory.
- 2 - to calculate and view the total inventory value.
- 3 - to exit the program.

For Choice 1 (Add Product):

The next input line is the string representing the product name as a string (single or multi-word, without quotes).

The next line is a double value representing the price as a decimal value

The next line is an integer value representing the quantity as an integer

For Choices 2 and 3, no additional input is required

Output Format

The output displays the results of the commands as follows:

- For the addProduct command, the program should display "Product added to inventory."
- For choice 2, the program should display "Total inventory value [totalvalue].
"The total value should be displayed with one decimal place. If there is no product in the inventory, print the total as 0.0.
- For choice 3, the program should exit

If the choice is not 1, 2, or 3, then print "Invalid choice. Please select a valid option (1/2/3).".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1

Laptop

800.0

3

2

5

3

Output: Product added to inventory.

Total inventory value: \$2400.0

Invalid choice. Please select a valid option (1/2/3).

Answer

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

```
interface Inventory{
```

```
    void addProduct(String name, double price, int quantity);  
    double calculateTotalValue();
```

```
}
```

```
class Product
```

```
{
```

```
    private String name;  
    private double price;  
    private int quantity;
```

```
    public Product(String name, double price, int quantity)
```

```
{
```

```
        this.name = name;  
        this.price = price;  
        this.quantity = quantity;
```

```
}
```

```
    public double getValue()
```

```
{
```

```
        return price * quantity;
```

```
}
```

```
}
```

```
class SimpleInventory implements Inventory
```

```
{
```

```
    private Product[] products;
```

```
    private int count;
```

```
    public SimpleInventory(int capacity)
```

```
{
```

```
        products = new Product[capacity];
```

```
        count = 0;
```

```
}
```

```
    public void addProduct(String name, double price, int quantity)
```

```
{
```

```
        if (count < products.length)
```

```
{
```

```
    products[count++] = new Product(name, price, quantity);  
    System.out.println("Product added to inventory.");
```

```
} else
```

```
{
```

```
    System.out.println("Inventory full. Cannot add more products.");
```

```
}
```

```
}
```

```
    public double calculateTotalValue()
```

```
{
```

```
    double total = 0.0;  
    for (int i = 0; i < count; i++)
```

```
{
```

```
        total += products[i].getValue();
```

```
}
```

```
    return total;
```

```
}
```

```
}
```

```
public class Main {
```

```

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    Inventory inventory = new SimpleInventory(10);
    while (true) {
        int choice = scanner.nextInt();
        if (choice == 1) {
            scanner.nextLine();
            String productName = scanner.nextLine();
            double price = scanner.nextDouble();
            int quantity = scanner.nextInt();
            inventory.addProduct(productName, price, quantity);
        } else if (choice == 2) {
            double totalValue = inventory.calculateTotalValue();
            System.out.println("Total inventory value: $" + totalValue);
        } else if (choice == 3) {
            break;
        } else {
            System.out.println("Invalid choice. Please select a valid option
(1/2/3).");
        }
    }
    scanner.close();
}
}

```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Raj is curious about how old he is in the current year.

He has asked you to create a simple program that calculates a person's age based on their birth year. You decide to implement this functionality using the AgeCalculator interface and the HumanAgeCalculator class.

Note: The current year is 2024. Calculate the current age by using the formula: current year - birth year.

Input Format

The input consists of an integer representing the birth year.

Output Format

The output displays "You are X years old." where X is an integer representing the calculated age based on the entered birth year.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1934

Output: You are 90 years old.

Answer

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

```
interface AgeCalculator
```

```
{
```

```
    int calculateAge(int birthYear);
```

```
}
```

```
class HumanAgeCalculator implements AgeCalculator
```

```
{
```

```
    public int calculateAge(int birthYear)
```

```
{
```

```
        return 2024 - birthYear;
```

```
    }
```

```
}
```

```
class AgeCalculatorApp {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        AgeCalculator ageCalculator = new HumanAgeCalculator();  
  
        int birthYear = scanner.nextInt();  
        int age = ageCalculator.calculateAge(birthYear);  
  
        System.out.println("You are " + age + " years old.");  
    }  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 8_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 13

Section 1 : MCQ

1. What will be the output for the following code?

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

```
class Test {  
    public static void main(String[] args) {  
        try {  
            double balance = -500;  
            if (balance < 0) {  
                throw new NegativeBalanceException("Balance cannot be  
negative");  
            }  
        }  
    }  
}
```

```
} catch (NegativeBalanceException e) {  
    System.out.println("Error: " + e.getMessage());  
}  
}  
}
```

Answer

Error: Balance cannot be negative

Status : Correct

Marks : 1/1

2. What will happen if a checked custom exception is thrown inside a method without being caught or declared?

Answer

Compilation Error

Status : Correct

Marks : 1/1

3. What will be the output for the following code?

```
import java.io.*;
```

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

```
class Test {  
    public static void main(String[] args) {  
        try {  
            int temperature = 110;  
            if (temperature > 100) {  
                throw new TemperatureTooHighException("Temperature too  
high");  
            }  
        } catch (TemperatureTooHighException e) {
```

```
        System.out.println(e.getMessage());
    }
}
}
```

Answer

Temperature too high

Status : Correct

Marks : 1/1

4. What will be the output for the following code?

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

```
class Test {
    public static void main(String[] args) {
        try {
            int age = 15;
            if (age < 18) {
                throw new InvalidVotingAgeException("You are not eligible to
vote");
            }
            System.out.println("Eligible to vote");
        } catch (InvalidVotingAgeException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Answer

Exception occurred

Status : Wrong

Marks : 0/1

5. What will be the output of the following code?

```
class MyException extends Exception {  
    public MyException() {  
        super("Default Exception Message");  
    }  
}
```

```
class Test {  
    public static void main(String[] args) {  
        try {  
            throw new MyException();  
        } catch (MyException e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

Answer

Default Exception Message

Status : Correct

Marks : 1/1

6. What will be the output for the following code?

```
import java.io.*;
```

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

```
class Test {  
    public static void main(String[] args) {  
        try {  
            int age = -5;  
            if (age < 0) {  
                throw new NegativeAgeException("Age cannot be negative");  
            }  
        }  
    }  
}
```

```
    } catch (NegativeAgeException e) {  
        System.out.println(e.getMessage());  
    }  
}  
}
```

Answer

Age cannot be negative

Status : Correct

Marks : 1/1

7. what is the output of the following code?

```
class MyException extends Exception {  
    public MyException(String message) {  
        super(message);  
    }  
}  
  
class Test {  
    static void check() throws MyException {  
        throw new MyException("Custom Exception Occurred");  
    }  
  
    public static void main(String[] args) {  
        try {  
            check();  
        } catch (Exception e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

Answer

Custom Exception Occurred

Status : Correct

Marks : 1/1

8. What will be the output for the following code?

```
import java.io.*;
```

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

```
class Test {  
    public static void main(String[] args) {  
        try {  
            int age = 17;  
            if (age < 18) {  
                throw new UnderageException("Underage, cannot proceed");  
            }  
        } catch (UnderageException e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

Answer

Underage, cannot proceed

Status : Correct

Marks : 1/1

9. Which keyword is used to explicitly throw a custom exception?

Answer

throw

Status : Correct

Marks : 1/1

10. what is the output of the following code?

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

```

    }
}

class Test {
    public static void main(String[] args) {
        try {
            throw new MyException("Error occurred");
        } catch (MyException e) {
            System.out.println(e);
        }
    }
}

```

Answer

MyException: Error occurred

Status : Correct

Marks : 1/1

11. What will be the output for the following code?

```

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

class Test {
    public static void main(String[] args) {
        try {
            String username = "abc";
            if (username.length() < 5) {
                throw new InvalidUsernameException("Username must be at
least 5 characters long");
            }
        } catch (InvalidUsernameException e) {
            System.out.println(e.getMessage());
        }
    }
}

```

Answer

Username must be at least 5 characters long

Status : Correct

Marks : 1/1

12. What is the purpose of a custom exception in Java?

Answer

To create user-defined exceptions for specific scenarios

Status : Correct

Marks : 1/1

13. Which of the following is true about custom exceptions?

Answer

Custom exceptions must extend either Exception or RuntimeException

Status : Correct

Marks : 1/1

14. How do you create an unchecked custom exception?

Answer

By extending RuntimeException

Status : Correct

Marks : 1/1

15. What will be the output for the following code?

```
import java.io.*;
```

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

```
class Test {
```

```
public static void main(String[] args) {  
    try {  
        int stock = 0;  
        if (stock == 0) {  
            throw new OutOfStockException("Item is out of stock");  
        }  
    } catch (OutOfStockException e) {  
        System.out.println(e.getMessage());  
    }  
}
```

Answer

Compilation Error

Status : Wrong

Marks : 0/1

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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

DotException AtTheRateException DomainException

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

// You are using Java

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

```
class DotException extends Exception
```

```
{
```

```
    DotException(String message)
```

```
{
```

```
        super(message);
```

```
}
```

```
}
```

```
class AtTheRateException extends Exception
```

```
{
```

```
    AtTheRateException(String message)
```

```
{
```

```
        super(message);
```

```
}  
}
```

```
class DomainException extends Exception
```

```
{
```

```
    DomainException(String message)
```

```
{
```

```
    super(message);
```

```
}
```

```
}
```

```
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");
```

```
}
```

```
}
```

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

```
{
```

```
    int atCount = email.length() - email.replace("@", "").length();
    if (atCount != 1 || email.startsWith("@") || email.endsWith("@") ||
        email.contains("@@"))
```

```
{  
    throw new AtTheRateException("Invalid @ usage");  
}  
    if (email.endsWith(".") || email.startsWith(".") || email.contains(".."))  
    {  
        throw new DotException("Invalid Dot usage");  
    }  
    int atIndex = email.indexOf('@');  
    int lastDotIndex = email.lastIndexOf('.');  
    if (lastDotIndex < atIndex + 2 || lastDotIndex == email.length() - 1)  
    {  
        throw new DotException("Invalid Dot usage");  
    }  
    String domain = email.substring(lastDotIndex + 1);  
    if (!(domain.equals("com") || domain.equals("in") || domain.equals("net") ||  
domain.equals("biz")))  
    {  
        throw new DomainException("Invalid Domain");  
    }  
}
```

}

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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

```
import java.util.Scanner;

class InvalidDurationException extends Exception
{

    InvalidDurationException(String message)
{
```

```
        super(message);
```

```
    }
```

```
}
```

```
class ElsaMeetingScheduler
```

```
{
```

```
    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 class Main
```

```
{
```

```
    public static void main(String[] args)
```

```
{
```

```
Scanner sc = new Scanner(System.in);
int duration = sc.nextInt();
try

{

    ElsaMeetingScheduler.validateMeetingDuration(duration);
    System.out.println("Meeting scheduled successfully!");

} catch (InvalidDurationException e)
{

    System.out.println("Error: " + e.getMessage());

}

}

}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In a user registration system, there is a requirement to implement a username validation module. Users attempting to register must adhere to specific criteria for their usernames to be considered valid.

Your task is to develop a program that takes user input for a desired username and validates it according to the following rules:

The username must not contain any spaces. The username must be at least 5 characters long.

Implement a custom exception, `InvalidUsernameException`, to handle cases where the entered username does not meet the specified criteria.

Input Format

The input consists of a string S, representing the desired username.

Output Format

If the username is valid, print "Username is valid: [S]".

If the username is invalid:

1. If the username is short, print "Invalid Username: Username must be at least 5 characters long"
2. If the username contains spaces, print "Invalid Username: Username cannot contain spaces"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: John

Output: Invalid Username: Username must be at least 5 characters long

Answer

// You are using Java

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

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

```
public class Main {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        String username = sc.nextLine();  
        try {  
            validateUsername(username);  
            System.out.println("Username is valid: " + username);  
        } catch (InvalidUsernameException e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

```
public static void validateUsername(String username) throws
InvalidUsernameException {
    if (username.contains(" ")) {
        throw new InvalidUsernameException("Invalid Username: Username
cannot contain spaces");
    } else if (username.length() < 5) {
        throw new InvalidUsernameException("Invalid Username: Username must
be at least 5 characters long");
    }
}
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

A local municipality is implementing an online voting system for a community event and wants to ensure that only eligible voters (those aged 18 or older) can participate.

Your task is to develop a program that validates the age of individuals attempting to vote online. If the user's age is below 18, the program should throw a custom exception, `InvalidAgeException`, preventing them from casting their vote. If the input is invalid, catch the appropriate `InputMismatchException` and print the in-built exception message.

Input Format

The input consists of an integer representing the age.

Output Format

If the age is 18 or older, print "Eligible to vote"

If the age is below 18, print "Exception occurred: InvalidAgeException: Age is not valid to vote"

If there is any other type of exception, print "An error occurred: " followed by the in-built exception message.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 20

Output: Eligible to vote

Answer

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

class InvalidAgeException extends Exception
{
    InvalidAgeException(String message)
    {
        super(message);
    }
}

public class Main
{
```

```
static void validateAge(int age) throws InvalidAgeException
{

    if (age < 18)
    {

        throw new InvalidAgeException("Age is not valid to vote");

    }

}

public static void main(String[] args)
{

    Scanner sc = new Scanner(System.in);
    try
    {

        int age = sc.nextInt();
        validateAge(age);
        System.out.println("Eligible to vote");

    } catch (InvalidAgeException e)
    {

        System.out.println("Exception occurred: InvalidAgeException: " +
e.getMessage());
    }
}
```

```
} catch (InputMismatchException e)
```

```
{
```

```
    System.out.println("An error occurred: " + e);
```

```
}
```

```
}
```

```
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In a file management system, users are required to provide a valid file name when creating new files. The system enforces specific rules for file names to maintain consistency and avoid potential issues. Your task is to implement a Java program named `FileNameValidator` that takes user input for a file name and validates it according to the specified rules.

Rules for Valid File Name:

The file name must consist of alphanumeric characters (letters and digits) only. The file name must have a minimum length of 3 characters.

Implement a custom exception, `FileNameValidator`, to handle cases where the entered filename does not meet the specified criteria.

Input Format

The input consists of a string S, representing the desired filename.

Output Format

The output is displayed in the following format:

If the entered file name meets the specified criteria, the program outputs

"Valid file name"

If the entered file name does not meet the criteria and triggers the `InvalidFileNameException`, the program outputs

"Error: Invalid file name. It must be alphanumeric and have a minimum length of 3 characters."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: myfile123

Output: Valid file name

Answer

```
// You are using Java
import java.util.Scanner;
class InvalidFileNameException extends Exception{
    InvalidFileNameException(String message){
        super(message);
    }
}
public class Main{
    static void validateFileName(String filename) throws
    InvalidFileNameException{
        if (filename.length() < 3 || !filename.matches("[a-zA-Z0-9]+")){
            throw new InvalidFileNameException("Invalid file name. It must be
            alphanumeric and have a minimum length of 3 characters.");
        }
    }
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        String filename = sc.nextLine();
        try{
```

```
        validateFileName(filename);  
        System.out.println("Valid file name");  
    } catch (InvalidFileNameException e){  
        System.out.println("Error: " + e.getMessage());  
    }  
}  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 9_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 12

Section 1 : MCQ

1. What will be the output of the following code?

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<>();
        list.add(10);
        list.add(20);
        list.add(30);
        list.remove(1);
        System.out.println(list);
    }
}
```

Answer

[10, 30]

Status : Correct

Marks : 1/1

2. What will be the output of the following code?

```
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<>();
        list.add(10);
        list.add(20);
        list.add(30);
        System.out.println("Size of the list: " + list.size());
    }
}
```

Answer

Size of the list: 3

Status : Correct

Marks : 1/1

3. What will be the output of the following code?

```
import java.util.*;

class Main {
    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<>();
        list.add(1);
        list.add(2);
        list.add(3);
        list.add(4);
        list.add(5);
        System.out.println(list.get(3));
    }
}
```

Answer

4

Status : Correct

Marks : 1/1

4. How can you access the first element of an ArrayList named as list?

Answer

list.get(0);

Status : Correct

Marks : 1/1

5. What will be the output of the following code?

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Stack<Integer> s = new Stack<>();
        s.push(10);
        s.push(20);
        s.push(30);
        System.out.println(s.peek());
    }
}
```

Answer

30

Status : Correct

Marks : 1/1

6. What will be the output of the following code?

```
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("Apple");
        list.add("Banana");
        list.remove("Apple");
        System.out.println(list);
    }
}
```

Answer

[Banana]

Status : Correct

Marks : 1/1

7. Which method is used to add an element to the top of the stack?

Answer

push()

Status : Correct

Marks : 1/1

8. What will be the output of the following code?

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<>();
        list.add(1);
        list.add(2);
        list.add(3);
        list.add(4);
        list.set(2, 10);
        System.out.println(list);
    }
}
```

Answer

[1, 10, 3, 4]

Status : Wrong

Marks : 0/1

9. What will be the output of the following code?

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Stack<Integer> stack = new Stack<>();
        for (int i = 1; i <= 3; i++)
            stack.push(i * 2);
        stack.pop();
        stack.push(10);
        System.out.println(stack.peek());
    }
}
```

Answer

10

Status : Correct

Marks : 1/1

10. What is Collection in Java?

Answer

A group of objects

Status : Correct

Marks : 1/1

11. Which of the following methods removes and returns the last element from a LinkedList?

Answer

pop()

Status : Wrong

Marks : 0/1

12. What is the correct way to create an ArrayList in Java?

Answer

ArrayList<String> list = new ArrayList<>();

Status : Correct

Marks : 1/1

13. What will be the output of the following code?

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("Java");
        list.add("Python");
        list.add("Java");
        list.add("C++");
        System.out.println(list.indexOf("Java"));
    }
}
```

Answer

0

Status : Correct

Marks : 1/1

14. What will be the output of the following code?

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("apple");
        list.add("banana");
        list.add("cherry");
        list.add("banana");
        System.out.println(list.lastIndexOf("banana"));
    }
}
```

Answer

3

Status : Correct

Marks : 1/1

15. What does the addFirst() method of LinkedList do?

Answer

Adds an element to the end of the list

Status : Wrong

Marks : 0/1

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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() || num > list.get(list.size() - 1)) {
                list.add(num);
            }
        }
        System.out.println(list);
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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

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

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

            if (command.equals("ADD")) {
                String song = sc.next();
                playlist.add(song);
                if (playlist.size() == 1) {
                    currentIndex = 0;
                }
            } else if (command.equals("REMOVE")) {
                String song = sc.next();
                int removeIndex = playlist.indexOf(song);
                if (removeIndex != -1) {
                    playlist.remove(removeIndex);
                    if (playlist.isEmpty()) {
                        currentIndex = 0;
                    } else if (removeIndex < currentIndex) {
                        currentIndex--;
                    } else if (currentIndex >= playlist.size()) {
                        currentIndex = 0;
                    }
                }
            } 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) + " ");
    }
}
}
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 9_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Assist Pranitha in developing a program that takes an integer N as input, representing the number of names to be read. Then read N names and store them in an ArrayList. Finally, input a search string and output the frequency of that string in the list of names.

Note: Some parts of the code are provided as snippets, and you need to complete the remaining sections by writing the necessary code.

Input Format

The first line of input consists of an integer N, representing the number of names to be read.

The following N lines consist of N names, as a string.

The last line consists of a string, representing the name to be searched.

Output Format

The output prints a single integer, representing the frequency of the specified name in the given list.

If the specified name is not found, print 0.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

Alice

Bob

Ankit

Alice

Pranitha

Alice

Output: 2

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<String> names = new ArrayList<>();  
        for (int i = 0; i < N; i++) {  
            names.add(sc.next());  
        }  
        String searchName = sc.next();  
        int count = 0;  
        for (String name : names) {  
            if (name.equals(searchName)) {  
                count++;  
            }  
        }  
    }  
}
```

```
}  
    System.out.print(count);  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 10_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 13

Section 1 : MCQ

1. Which method removes all elements from a Set?

Answer

clear()

Status : Correct

Marks : 1/1

2. What happens when you add duplicate elements to a HashSet?

Answer

The duplicate is ignored

Status : Correct

Marks : 1/1

3. Which of the following is true about TreeMap?

Answer

It maintains natural ordering

Status : Correct

Marks : 1/1

4. What will happen if you add a null element to a TreeSet?

Answer

An exception occurs

Status : Correct

Marks : 1/1

5. What will be the output of the following code?

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        HashMap<String, String> map = new HashMap<>();
        map.put("A", "Apple");
        map.put("B", "Banana");
        map.put("C", "Cherry");
        map.replace("B", "Blueberry");
        System.out.println(map);
    }
}
```

Answer

{A=Apple, B=Blueberry, C=Cherry}

Status : Correct

Marks : 1/1

6. What will happen if you add elements in descending order in a TreeSet?

Answer

They are sorted in ascending order

Status : Correct

Marks : 1/1

7. What will be the output of the following code?

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();
        map.put("A", 1);
        map.put("B", 2);
        map.put("C", 3);
        System.out.println(map.containsKey("B"));
    }
}
```

Answer

true

Status : Correct

Marks : 1/1

8. Which of the following allows null keys in Java?

Answer

HashMap

Status : Correct

Marks : 1/1

9. What is the time complexity of retrieving an element from a HashSet?

Answer

O(1)

Status : Correct

Marks : 1/1

10. Which statement is true about HashSet and TreeSet?

Answer

TreeSet provides sorted elements

Status : Correct

Marks : 1/1

11. How does HashSet check for duplicate elements?

Answer

Using equals() and hashCode()

Status : Correct

Marks : 1/1

12. What happens if two keys have the same hash code in a HashMap?

Answer

A linked list is used to store values with the same hash

Status : Correct

Marks : 1/1

13. Which method retrieves the lowest key in a TreeMap?

Answer

firstKey()

Status : Correct

Marks : 1/1

14. What will be the output of the following code?

```
import java.util.*;
class Main {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();
        map.put("X", 10);
        map.put("Y", 20);
        map.put("Z", 30);
        map.remove("Y");
        System.out.println(map);
    }
}
```

}

Answer

{X=10, Y=20, Z=30}

Status : Wrong

Marks : 0/1

15. Which of the following is true about HashMap?

Answer

It maintains the order of insertion

Status : Wrong

Marks : 0/1

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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 ma{  
    public static void main(String args[]){  
        Scanner jo=new Scanner(System.in);  
        int a=jo.nextInt();  
        HashSet<String> k=new HashSet<>();  
        jo.nextLine();  
        for(int i=0;i<a;i++){
```

```
k.add(jo.nextLine());  
}  
for(String ka:k){  
    System.out.println(ka);  
}  
}  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        HashMap<String, Double> map = new HashMap<>();
        double sum = 0;
        while (true) {
            String input = sc.next();
            if (input.equals("done")) break;
            if (!input.contains(":")) {
                System.out.print("Invalid format");
                return;
            }
            String[] parts = input.split(":");
            if (parts.length != 2) {
                System.out.print("Invalid format");
                return;
            }
            String fruit = parts[0];
            String qtyStr = parts[1];
            if (!fruit.matches("[A-Za-z]+")) {
                System.out.print("Invalid format");
            }
        }
    }
}
```

```
        return;
    }
    double qty;
    try {
        qty = Double.parseDouble(qtyStr);
    } catch (Exception e) {
        System.out.print("Invalid input");
        return;
    }
    map.put(fruit, qty);
}
for (double v : map.values()) sum += v;
System.out.printf("%.2f", sum);
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 10_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : COD

1. Problem Statement

Priya is analyzing encrypted messages in a research project. She wants to analyze the frequency of each character in a given paragraph. The characters should be stored in a TreeMap so that the output is sorted in ascending order of characters automatically.

You are required to build a Java program that:

Uses a `TreeMap<Character, Integer>` to count how many times each character appears in the message. Ignores spaces and considers only alphabets (case-sensitive). Outputs the frequencies of characters in sorted order.

You must use a TreeMap in the class named MessageAnalyzer.

Input Format

The first line of input contains an integer n, the number of lines in the message.

The next n lines each contain a string (the encrypted message line).

Output Format

The first line of output prints: "Character Frequency:"

Then print each character and its frequency in the format: "<character>: <count>"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 2
Hello World
Java

Output: Character Frequency:

H: 1

J: 1

W: 1

a: 2

d: 1

e: 1

l: 3

o: 2

r: 1

v: 1

Answer

// You are using Java

```
import java.util.*;
```

```
class MessageAnalyzer {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        int n = sc.nextInt();
```

```
        sc.nextLine();
```

```
        TreeMap<Character, Integer> map = new TreeMap<>();
```

```
        for (int i = 0; i < n; i++) {
```

```
            String line = sc.nextLine();
```

```
            for (char c : line.toCharArray()) {
```

```
        if (c != ' ' && Character.isLetter(c)) {  
            map.put(c, map.getDefault(c, 0) + 1);  
        }  
    }  
    System.out.print("Character Frequency: ");  
    for (Map.Entry<Character, Integer> e : map.entrySet()) {  
        System.out.print(e.getKey() + ": " + e.getValue() + " ");  
    }  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 10_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : COD

1. Problem Statement

In a ticket reservation system, you store the available seat numbers in a TreeSet. Users input their desired seat number, and the program checks whether the chosen seat is available.

Using a TreeSet ensures quick and efficient verification of seat availability, ensuring a smooth and organized ticket booking process.

Input Format

The first line of input contains a single integer n , representing the number of available seats.

The second line contains n space-separated integers, representing the available seat numbers.

The third line contains an integer m, representing the seat number that needs to be searched.

Output Format

The output displays "[m] is present!" if the given seat is available. Otherwise, it displays "[m] is not present!"

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 4

2 4 5 6

5

Output: 5 is present!

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();

 TreeSet<Integer> seats = new TreeSet<>();

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

 seats.add(sc.nextInt());

 }

 int m = sc.nextInt();

 if (seats.contains(m))

 System.out.print(m + " is present!");

 else

 System.out.print(m + " is not present!");

 }

}

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_Week 12_Java_Lamba Expressions_MCQ

Attempt : 1
Total Mark : 10
Marks Obtained : 9

Section 1 : MCQ

1. What is the return type of a lambda expression in Java?

Answer

The return type is inferred from the context

Status : Correct

Marks : 1/1

2. What is a lambda expression in Java?

Answer

A way to define anonymous methods

Status : Correct

Marks : 1/1

3. Which functional interface in Java takes two arguments and returns a result?

Answer

BiFunction

Status : Correct

Marks : 1/1

4. Can a lambda expression have more than one parameter?

Answer

Yes, it can have multiple parameters

Status : Correct

Marks : 1/1

5. Which of the following is a valid lambda expression in Java?

Answer

(x) -> x * 2

Status : Wrong

Marks : 0/1

6. Can a lambda expression in Java have a body with multiple statements?

Answer

Yes, if the statements are enclosed in curly braces

Status : Correct

Marks : 1/1

7. What is the syntax for a basic lambda expression in Java?

Answer

(parameters) -> expression

Status : Correct

Marks : 1/1

8. Can a lambda expression in Java have a body with multiple statements?

Answer

Yes, if the statements are enclosed in curly braces

Status : Correct

Marks : 1/1

9. Which functional interface is commonly used with lambda expressions in Java?

Answer

Runnable

Status : Correct

Marks : 1/1

10. Which of the following interfaces is NOT a functional interface in Java?

Answer

Iterable

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 12_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Sabrina is working on a project that involves analyzing a set of numbers. In her exploration, she encounters scenarios where extracting even numbers and finding their sum is essential.

Create a program that calculates the sum of even numbers from a given array of integers using a lambda expression.

Input Format

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

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

Output Format

The output prints the sum of the even integers from the array.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 3
29 37 45
Output: 0

Answer

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

public 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();
        }
        int sum = Arrays.stream(arr).filter(x -> x % 2 == 0).sum();
        System.out.print(sum);
        sc.close();
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 12_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Alex is learning about Java's functional interfaces and lambda expressions.

He wants to write a simple program that prints the square of each number in an array using a predefined functional interface.

Help Alex complete this task using the Consumer functional interface.

Input Format

- The first line contains an integer N, the number of elements in the array.
- The second line contains N space-separated integers.

Output Format

- Print the squares of all elements in the array, separated by a space.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 4

1 2 3 4

Output: 1 4 9 16

Answer

// You are using Java

import java.util.*;

import java.util.function.*;

```
public 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();  
        }  
        Consumer<Integer> printSquare = x -> System.out.print((x * x) + " ");  
        for (int num : arr) {  
            printSquare.accept(num);  
        }  
        sc.close();  
    }  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 12_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In the mystical realm of programming, there exists a magical incantation to reveal hidden words.

Elara, the skilled enchantress, wishes to summon a word using her spell and then reverse its characters to uncover its enchanted reflection.

Write a program that uses the predefined functional interface `Supplier<String>` and a lambda expression to:

Supply (generate) a string, and

Display its reversed form.

Input Format

No input is required from the user.

The string must be supplied internally using a Supplier<String>.

Output Format

Print the reversed version of the supplied string.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Wizard!!

Output: !!draziW

Answer

```
import java.util.*;
import java.util.function.*;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String input = sc.nextLine();
        Supplier<String> supplier = () -> input;
        String str = supplier.get();
        String reversed = new StringBuilder(str).reverse().toString();
        System.out.print(reversed + " ");
        sc.close();
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 12_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Abi is working on a text analysis project where she needs to categorize words based on their length.

Words that have three or fewer characters are considered "Short", while words with more than three characters are classified as "Long."

Write a Java program that takes a sentence as input, analyzes each word, and prints a list showing whether each word is "Short" or "Long."

Use the predefined functional interface `Function<String, String>` along with a lambda expression for categorization.

Input Format

A single line containing a sentence (words separated by spaces).

Output Format

- A single line with each word categorized as "Short" or "Long", separated by spaces.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: I love my cat

Output: Short Long Short Short

Answer

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

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String sentence = sc.nextLine();
        String[] words = sentence.split(" ");
        Function<String, String> categorize = word -> word.length() <= 3 ? "Short" :
"Long";
        for (String word : words) {
            System.out.print(categorize.apply(word) + " ");
        }
        sc.close();
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: sabariprabu M
Email: 240701446@rajalakshmi.edu.in
Roll no: 2116240701446
Phone: 6369800633
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 11

Attempt : 1
Total Mark : 20
Marks Obtained : 20

Section 1 : Project

1. Problem Statement

In Café Central, the menu is cataloged and stored in a database.

To efficiently manage the restaurant's menu using Java and JDBC, you must build a Restaurant Management System that supports:

Adding new menu items

Updating menu item prices

Viewing details of a menu item

Displaying all menu items in sorted order

You are given two files:

File 1: MenuItem.java (POJO Class)

This class represents the MenuItem entity.

A MenuItem contains the following details:

Field	Description
itemId	Unique Menu Item ID (Integer)
name	Item Name (String)
category	Item Category (String)
price	Item Price (Double)

Students must write code in the marked area:

```
class MenuItem {  
    private int itemId;  
    private String name;  
    private String category;  
    private double price;  
  
    public MenuItem() {}  
  
    public MenuItem(int itemId, String name, String category, double price) {  
        // write your code here  
    }  
  
    // Include getters and setters  
}
```

Expected in this part:

Assign parameter values to instance variables inside the constructor.

Add getters and setters for all attributes.

File 2: MenuItemDAO.java (Data Access Layer)

This class handles all database operations using JDBC.

Students must complete the missing JDBC logic in the following methods:

```
class MenuItemDAO {

    public void addItem(Connection conn, MenuItem menuItem)
    throws SQLException {

        // write your code here

    }

    public void updateItemPrice(Connection conn, int itemId, double
    newPrice) throws SQLException {

        // write your code here

    }

    public void deleteMenuItem(Connection conn, int itemId) throws
    SQLException {

        // write your code here

    }

    public MenuItem viewItemDetails(Connection conn, int itemId) throws
    SQLException {

        // write your code here

    }

    public List<MenuItem> displayAllMenuItems(Connection conn) throws
    SQLException {

        // write your code here

    }

    private MenuItem mapToMenuItem(ResultSet rs) throws SQLException {
        return new MenuItem(
```

```
        // write your code here
    );
}
}
```

Expected in this part:

Write SQL queries for INSERT, UPDATE, DELETE, SELECT.

Execute queries using PreparedStatement or Statement.

Map ResultSet rows to MenuItem objects using mapToMenuItem().

Return a List<MenuItem> where required.

The system should connect to a MySQL database using the following default credentials:

DB URL: jdbc:mysql://localhost/ri_db

USER: test

PWD: test123

The menu table has already been created with the following structure:

Table Name: menu

Input Format

The first line of input consists of an integer choice, representing the operation to be performed (1 for Add Item, 2 for Restock item, 3 for reduce item, 4 for Display, 5 for Exit).

For choice 1 (Add Menu Item):

- The second line consists of an integer item_id.
- The third line consists of a string name.
- The fourth line consists of a string category.
- The fifth line consists of a double price.

For choice 2 (Update Item Price):

- The second line consists of an integer item_id.
- The third line consists of a double new_price.

For choice 3 (View Item Details):

- The second line consists of an integer item_id.

For choice 4 (Display All Menu Items):

- No additional inputs are required.

For choice 5 (Exit):

- No additional inputs are required.

Output Format

For choice 1 (Add Menu Item):

- Print "Menu item added successfully" if the item was added.
- Print "Failed to add item." if the insertion failed.

For choice 2 (Update Item Price):

- Print "Item price updated successfully" if the price update was successful.
- Print "Item not found." if the specified item ID does not exist.

For choice 3 (View Item Details):

- Display the item details in the format:
- ID: [item_id] | Name: [name] | Category: [category] | Price: [price]
- Print "Item not found." if the specified item ID does not exist.

For choice 4 (Display All Menu Items):

- Display each item on a new line in the format:
- ID | Name | Category | Price
- If no items are available, print nothing (or handle with an appropriate message if desired).

For choice 5 (Exit):

- Print "Exiting Restaurant Management System."

For invalid input:

- Print "Invalid choice. Please try again."

Sample Test Case

Input: 1

11

Margherita Pizza

Main Course

12.99

4

5

Output: Menu item added successfully

ID | Name | Category | Price

11 | Margherita Pizza | Main Course | 12.99

Exiting Restaurant Management System.

Answer

```
import java.sql.*;
```

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

```
class RestaurantManagementSystem {
```

```
    public static void main(String[] args) {
```

```
        try (Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/ri_db", "test", "test123");
```

```
            Scanner scanner = new Scanner(System.in)) {
```

```
            boolean running = true;
```

```
            while (running) {
```

```
                int choice = scanner.nextInt();
```

```
                switch (choice) {
```

```
                    case 1:
```

```
                        addMenuItem(conn, scanner);
```

```
                        break;
```

```
                    case 2:
```

```
                        updateItemPrice(conn, scanner);
```

```
                        break;
```

```

        case 3:
            viewItemDetails(conn, scanner);
            break;
        case 4:
            displayAllMenuItems(conn);
            break;
        case 5:
            System.out.println("Exiting Restaurant Management System.");
            running = false;
            break;
        default:
            System.out.println("Invalid choice. Please try again.");
    }
}
} catch (SQLException e) {
    e.printStackTrace();
}
}

```

```

public static void addItem(Connection conn, Scanner scanner) {
    int itemId = scanner.nextInt();
    scanner.nextLine(); // Consume newline

```

```

    String name = scanner.nextLine();
    String category = scanner.nextLine();
    double price = scanner.nextDouble();

```

```

    MenuItem menuItem = new MenuItem(itemId, name, category, price); //
    Using POJO

```

```

    String insertQuery = "INSERT INTO menu (item_id, name, category, price)
VALUES (?, ?, ?, ?)";

```

```

    try (PreparedStatement stmt = conn.prepareStatement(insertQuery)) {
        stmt.setInt(1, menuItem.getItemId());
        stmt.setString(2, menuItem.getName());
        stmt.setString(3, menuItem.getCategory());
        stmt.setDouble(4, menuItem.getPrice());
    }

```

```

    int rowsInserted = stmt.executeUpdate();
    System.out.println(rowsInserted > 0 ? "Menu item added successfully" :
"Failed to add item.");
} catch (SQLException e) {
    System.out.println("Error adding item: " + e.getMessage());
}
}

```

```

    }
}

public static void updateItemPrice(Connection conn, Scanner scanner) {
    int itemId = scanner.nextInt();
    double newPrice = scanner.nextDouble();

    String updateQuery = "UPDATE menu SET price = ? WHERE item_id = ?";
    try (PreparedStatement stmt = conn.prepareStatement(updateQuery)) {
        stmt.setDouble(1, newPrice);
        stmt.setInt(2, itemId);

        int rowsUpdated = stmt.executeUpdate();
        System.out.println(rowsUpdated > 0 ? "Item price updated successfully" :
"Item not found.");
    } catch (SQLException e) {
        System.out.println("Error updating price: " + e.getMessage());
    }
}

```

```

public static void viewItemDetails(Connection conn, Scanner scanner) {
    int itemId = scanner.nextInt();

    String selectQuery = "SELECT * FROM menu WHERE item_id = ?";
    try (PreparedStatement stmt = conn.prepareStatement(selectQuery)) {
        stmt.setInt(1, itemId);

        ResultSet rs = stmt.executeQuery();
        if (rs.next()) {
            MenuItem menuItem = new MenuItem(
                rs.getInt("item_id"),
                rs.getString("name"),
                rs.getString("category"),
                rs.getDouble("price")
            );

            System.out.printf("ID: %d | Name: %s | Category: %s | Price: %.2f%n",
                menuItem.getItemId(),
                menuItem.getName(),
                menuItem.getCategory(),
                menuItem.getPrice());
        } else {

```

```

        System.out.println("Item not found.");
    }
} catch (SQLException e) {
    System.out.println("Error retrieving item details: " + e.getMessage());
}
}

public static void displayAllMenuItems(Connection conn) {
    String displayQuery = "SELECT * FROM menu ORDER BY item_id";
    try (Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(displayQuery)) {

        System.out.println("ID | Name | Category | Price");
        while (rs.next()) {
            MenuItem menuItem = new MenuItem(
                rs.getInt("item_id"),
                rs.getString("name"),
                rs.getString("category"),
                rs.getDouble("price")
            );

            System.out.printf("%d | %s | %s | %.2f\n",
                menuItem.getItemId(),
                menuItem.getName(),
                menuItem.getCategory(),
                menuItem.getPrice());
        }
    } catch (SQLException e) {
        System.out.println("Error displaying menu items: " + e.getMessage());
    }
}
}

```

```

class MenuItem {
    private int itemId;
    private String name;
    private String category;
    private double price;

    // Constructor
    public MenuItem(int itemId, String name, String category, double price) {
        this.itemId = itemId;
    }
}

```

```
this.name = name;
this.category = category;
this.price = price;
}

// Getters and Setters
public int getItemId() { return itemId; }
public void setItemId(int itemId) { this.itemId = itemId; }

public String getName() { return name; }
public void setName(String name) { this.name = name; }

public String getCategory() { return category; }
public void setCategory(String category) { this.category = category; }

public double getPrice() { return price; }
public void setPrice(double price) { this.price = price; }
}

//
```

Status : Correct

Marks : 10/10

2. Problem Statement

Create a JDBC-based School Management System that handles runtime input to manage student records. The system should allow users to:

Add a new student (student ID, name, grade level, GPA).

Update a student's GPA, ensuring the GPA value is within the valid range (0.0 - 4.0).

View a specific student's record by student ID.

Display all students in the database.

Exit the application.

The system should connect to a MySQL database using the following default credentials:

DB URL: jdbc:mysql://localhost/ri_db

USER: test

PWD: test123

The students table has already been created with the following structure:

Table Name: students

Input Format

The first line of input consists of an integer choice, representing the operation to be performed:

(1 for Add Student, 2 for Update GPA, 3 for View Student Record, 4 for Display All Students, 5 for Exit)

For choice 1 (Add Student):

- The second line consists of an integer student_id.
- The third line consists of a string name.
- The fourth line consists of a string grade_level.
- The fifth line consists of a double gpa (must be between 0.0 and 4.0).

For choice 2 (Update GPA):

- The second line consists of an integer student_id.
- The third line consists of a double new_gpa (must be between 0.0 and 4.0).

For choice 3 (View Student Record):

- The second line consists of an integer student_id.

For choice 4 (Display All Students):

- No additional inputs are required.

For choice 5 (Exit):

- No additional inputs are required.

Output Format

The output displays:

For choice 1 (Add Student):

- Print "Student added successfully" if the student was added.
- Print "Failed to add student." if the insertion failed.

For choice 2 (Update GPA):

- Print "GPA updated successfully" if the GPA update was successful.
- Print "Student not found." if the specified student ID does not exist.
- Print "GPA must be between 0.0 and 4.0." if the provided GPA is out of the valid range.

For choice 3 (View Student Record):

- Display the student details in the format:
- ID: [student_id] | Name: [name] | Grade Level: [grade_level] | GPA: [gpa]
- Print "Student not found." if the specified student ID does not exist.

For choice 4 (Display All Students):

- Display each student on a new line in the format:
- ID | Name | Grade Level | GPA
- If there are no records, print nothing (or handle with an appropriate message if desired).

For choice 5 (Exit):

- Print "Exiting School Management System."

For invalid input:

- Print "Invalid choice. Please try again."

Sample Test Case

Input: 1

101

Alice Johnson

10

3.8

5

Output: Student added successfully
Exiting School Management System.

Answer

```
import java.sql.*;
import java.util.Scanner;

class SchoolManagementSystem {
    public static void main(String[] args) {
        try (Connection conn = DriverManager.getConnection("jdbc:mysql://
localhost/ri_db", "test", "test123"));
        Scanner scanner = new Scanner(System.in)) {

            boolean running = true;

            while (running) {

                int choice = scanner.nextInt();

                switch (choice) {
                    case 1:
                        addStudent(conn, scanner);
                        break;
                    case 2:
                        updateGrades(conn, scanner);
                        break;
                    case 3:
                        viewStudentRecord(conn, scanner);
                        break;
                    case 4:
                        displayAllStudents(conn);
                        break;
                    case 5:
                        System.out.println("Exiting School Management System.");
                        running = false;
                        break;
                    default:
                        System.out.println("Invalid choice. Please try again.");
                }
            }
        }
    }
}
```

```

    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public static void addStudent(Connection conn, Scanner scanner) {
    int studentId = scanner.nextInt();
    scanner.nextLine(); // Consume newline

    String name = scanner.nextLine();

    String gradeLevel = scanner.nextLine();

    double gpa = scanner.nextDouble();

    String insertQuery = "INSERT INTO students (student_id, name, grade_level,
gpa) VALUES (?, ?, ?, ?)";
    try (PreparedStatement stmt = conn.prepareStatement(insertQuery)) {
        stmt.setInt(1, studentId);
        stmt.setString(2, name);
        stmt.setString(3, gradeLevel);
        stmt.setDouble(4, gpa);

        int rowsInserted = stmt.executeUpdate();
        System.out.println(rowsInserted > 0 ? "Student added successfully" :
"Failed to add student.");
    } catch (SQLException e) {
        System.out.println("Error adding student: " + e.getMessage());
    }
}

public static void updateGrades(Connection conn, Scanner scanner) {
    int studentId = scanner.nextInt();

    double newGpa = scanner.nextDouble();

    // Validate GPA range
    if (newGpa < 0.0 || newGpa > 4.0) {
        System.out.println("GPA must be between 0.0 and 4.0.");
        return;
    }

    String updateQuery = "UPDATE students SET gpa = ? WHERE student_id
= ?";

```

```

        try (PreparedStatement stmt = conn.prepareStatement(updateQuery)) {
            stmt.setDouble(1, newGpa);
            stmt.setInt(2, studentId);

            int rowsUpdated = stmt.executeUpdate();
            System.out.println(rowsUpdated > 0 ? "GPA updated successfully" :
"Student not found.");
        } catch (SQLException e) {
            System.out.println("Error updating GPA: " + e.getMessage());
        }
    }
}

```

```

public static void viewStudentRecord(Connection conn, Scanner scanner) {
    int studentId = scanner.nextInt();

```

```

    String selectQuery = "SELECT * FROM students WHERE student_id = ?";
    try (PreparedStatement stmt = conn.prepareStatement(selectQuery)) {
        stmt.setInt(1, studentId);

```

```

        ResultSet rs = stmt.executeQuery();
        if (rs.next()) {
            System.out.printf("ID: %d | Name: %s | Grade Level: %s | GPA: %.2f%n",
                rs.getInt("student_id"),
                rs.getString("name"),
                rs.getString("grade_level"),
                rs.getDouble("gpa"));
        } else {

```

```

            System.out.println("Student not found.");
        }
    } catch (SQLException e) {
        System.out.println("Error retrieving student record: " + e.getMessage());
    }
}

```

```

public static void displayAllStudents(Connection conn) {
    String displayQuery = "SELECT * FROM students ";
    try (Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(displayQuery)) {

        System.out.println("ID | Name | Grade Level | GPA");
        while (rs.next()) {
            System.out.printf("%d | %s | %s | %.2f%n",

```

```
        rs.getInt("student_id"),
        rs.getString("name"),
        rs.getString("grade_level"),
        rs.getDouble("gpa"));
    }
} catch (SQLException e) {
    System.out.println("Error displaying students: " + e.getMessage());
}
}
```

Status : Correct

Marks : 10/10