

NAME: DEEPIKA A

CLASS: III CSE A

ROLL NO: 21CS029

DATE: 18/09/2023

### JAVA BASIC PROBLEMS

1) From the following details generate a report showing the cash position at the end of a month for aman.

- a) royalty received = rs 12000 ,
- b) machinery maintenance = rs 4000 ,
- c) sales commission = rs 3000 ,
- d) interest on fixed deposit : rs 3000 ,
- e) maintainence charges = rs 4000 ,
- f) property tax = rs 2000 ,
- g) bonus = rs 8000 ,
- h) gratuity = rs 5000 .

Find out whether the aman have surplus amount or deficit amount.

#### **CODE:**

```
import java.util.Scanner;

public class CashPosition {

    public static void main(String[] args) {

        // Initialize the cash position

        int cashPosition = 12000 - 4000 + 3000 + 3000 - 4000 - 2000 + 8000 - 5000;

        // Determine whether Aman has a surplus amount or deficit amount

        String cashPositionStatus;

        if (cashPosition >= 0) {

            cashPositionStatus = "Surplus";

        } else {

            cashPositionStatus = "Deficit";

        }

        // Print the cash position report
```

```

        System.out.println("Cash position at the end of the month: " + cashPosition);
        System.out.println("Cash position status: " + cashPositionStatus);
    }
}

```

2)Mr.M of Ajarbaijan wants to automate his bank details with opening. Account balance of rs 2,00,000

a)Deposit= rs 1,00,000

b)funds transfer to s = rs 25,000

c)loan taken from bank @repaid = rs 10,000

d)interest on the loan @5% for 6 months @S.I(simple interest)

e)withdrawal = rs 20,000

Automate his account balance through a programming code using java.

### **CODE:**

```

public class BankAccountAutomation {
    public static void main(String[] args) {
        double accountBalance = 200000.0; // Initial account balance
        // Transactions
        double deposit = 100000.0;
        double fundsTransfer = 25000.0;
        double loanTaken = 10000.0;
        double interestRate = 0.05; // 5% interest rate
        int loanDurationMonths = 6;
        double withdrawal = 20000.0;
        // Apply transactions
        accountBalance += deposit;
        accountBalance -= fundsTransfer;
        accountBalance += loanTaken;
        // Calculate interest on the loan (Simple Interest)
        double interest = (loanTaken * interestRate * loanDurationMonths) / 12;
        accountBalance -= interest;
    }
}

```

```

        accountBalance -= withdrawal;

        // Display the final account balance

        System.out.println("Mr. M's Final Account Balance: Rs " + accountBalance);
    }
}

```

3)Mr B and Mr U are friends who have their residences at opposite lanes with door no 10 and 20 respectively. During a rainy they missed their lanes and U reaches B's home and B reached U's home. They want to find their perfect ways. kindly guide them with a programming code using java.

**CODE:**

```

public class HomeDirections {

    public static void main(String[] args) {

        int bHome = 10; // Mr. B's home address

        int uHome = 20; // Mr. U's home address

        // Swap their locations

        int temp = bHome;

        bHome = uHome;

        uHome = temp;

        System.out.println("Mr. B, please go to Door No. " + bHome);

        System.out.println("Mr. U, please go to Door No. " + uHome);

    }

}

```

4)Ms.Sruti wants to detect the patterns for calendar year. Her focus is on the first even month of a year and its nature based on which she wants to plan certain things. The no.of days is of more significance in her view point.How will help sruthi through a programming code using java.(LEAP year logic)

**CODE:**

```

import java.util.Scanner;

public class LeapYearChecker {

```

```

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter a year: ");
    int year = scanner.nextInt();

    if (isLeapYear(year)) {
        System.out.println(year + " is a leap year.");
    } else {
        System.out.println(year + " is not a leap year.");
    }
    scanner.close();
}

public static boolean isLeapYear(int year) {
    return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
}
}

```

5)Deepak chawla classifies number as strong and weak numbers. A number is strong when it is divided by a minimum even value and does not produce a remainder. if it produces a remainder value it is considered to be a weak number. Help deepak in coding this logic using java (odd or even logic)

**CODE:**

```

import java.util.Scanner;

public class odd_even_classifier {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();
        if (isStrongNumber(number)) {
            System.out.println(number + " is a strong number.");
        }
    }
}

```

```

    } else {
        System.out.println(number + " is a weak number.");
    }
    scanner.close();
}

public static boolean isStrongNumber(int number) {
    if (number < 2) {
        return false; // Numbers less than 2 are neither strong nor weak
    }
    // The minimum even value = 2 because 0 is a neutral number
    return number % 2 == 0; // Check if the number is divisible by 2
}
}

```

6)MS edex is a vaccine manufacturing company and wants to prepare a report in java with following details below annual production of vaccine according to market is 20000 units and the cost of making the vaccine is rs 20 per bottle and storage cost is the rs 10 per bottle find out the optimum quantity of vaccine that would be produced without loss after calculation make the necessary adjustments to customize the report

a)If the making cost is decreased by 20% what will be the quantity?

b)If the storing cost is increased by 10% what will be the cost?

hint- formula for calculating optimum quantity

optimum quantity= overall square root ( $2 \times \text{annual production} \times \text{making cost} / \text{storing cost}$ )

#### **CODE:**

```

class vaccine{
    public static void main(String[] args) {
        double annual=20000;
        double mancost=20;
    }
}

```

```

double storage=10;
double val=(2*annual*mancost)/storage;
System.out.println("original value "+Math.sqrt(val));
double mancost1=mancost-(20*mancost/100);
double val1=(2*annual*mancost1)/storage;
System.out.println("mc decreases 20% value "+Math.sqrt(val1));
double storage1=storage+(10*storage/100);
double val2=(2*annual*mancost)/storage1;
System.out.println("stor increases 10% value "+Math.sqrt(val2));
}
}

```

7)Almitha industries is a steel manufacturing unit it had a machinery which was purchased for 10 lakhs and was used for 25 years . The machinery got damaged and was sold for 2 lakhs 50 thousand which is a scrap value. Write a java program to calculate the depreciation rate.

#### **CODE:**

```

public class DepreciationCalculator {
    public static void main(String[] args) {
        double initialCost = 1000000.0; // Initial cost in lakhs
        double scrapValue = 250000.0; // Scrap value in lakhs
        int usefulLife = 25; // Useful life in years
        // Calculate depreciation amount
        double depreciationAmount = (initialCost - scrapValue) / usefulLife;
        // Calculate depreciation rate
        double depreciationRate = (depreciationAmount / initialCost) * 100;
        System.out.println("Initial Cost: " + initialCost + " lakhs");
        System.out.println("Scrap Value: " + scrapValue + " lakhs");
        System.out.println("Useful Life: " + usefulLife + " years");
        System.out.println("Depreciation Amount per Year: " + depreciationAmount + " lakhs");
    }
}

```

```

        System.out.println("Depreciation Rate per Year: " + depreciationRate + "%");
    }
}

```

8) Primazon company limited for category 1 , product name :Garments, Quantity: 1500, rate:?,value: 12000 , for category 2, product name: Books, Quantity: ?,rate: 25, value: 10000 ; for category 3 ,product name: electronic gadgets, quantity: 20000, rate:35, value:?

i) write the java program to find out the following information such as rate for category 1,quantity for category 2 and value for category 3.

ii) Write a Java program to calculate the maximum quantity in all the three categories.

iii)Average rate for all the three categories.

iv)Minimum value for all 3 categories.

### CODE

```

i) public class PrimazonCompany {
    public static void main(String[] args) {
        // Given values
        int category1Quantity = 1500;
        double category1Value = 12000;
        double category2Rate = 25;
        double category2Value = 10000;
        int category3Quantity = 20000;
        double category3Rate = 35;
        // Calculate the missing values
        double category1Rate = category1Value / category1Quantity;
        int category2Quantity = (int) (category2Value / category2Rate);
        double category3Value = category3Rate * category3Quantity;
        // Print the results
        System.out.println("Rate for Category 1 (Garments): " + category1Rate);
        System.out.println("Quantity for Category 2 (Books): " + category2Quantity);
        System.out.println("Value for Category 3 (Electronic Gadgets): " + category3Value);
    }
}

```

```
}  
}
```

```
ii) public class PrimazonCompany {  
    public static void main(String[] args) {  
        // Given values  
        int category1Quantity = 1500;  
        int category2Quantity = 0; // To be calculated  
        int category3Quantity = 20000;  
        // Calculate category 2 quantity based on the given value  
        double category2Value = 10000;  
        double category2Rate = 25;  
        category2Quantity = (int) (category2Value / category2Rate);  
        // Find the maximum quantity among all three categories  
        int maxQuantity = Math.max(category1Quantity, Math.max(category2Quantity,  
category3Quantity));  
        // Print the maximum quantity  
        System.out.println("Maximum Quantity among all three categories: " + maxQuantity);  
    }  
}
```

```
iii) public class PrimazonCompany {  
    public static void main(String[] args) {  
        // Given values  
        int category1Quantity = 1500;  
        double category1Value = 12000;  
        double category2Rate = 25;  
        double category2Value = 10000;  
        int category3Quantity = 20000;  
        double category3Rate = 35;  
        // Calculate the rates for each category  
        double category1Rate = category1Value / category1Quantity;
```



```

        double category2Rate = category2Value / category1Quantity;
        double category3Rate = category3Rate;
        // Calculate the average rate for all three categories
        double averageRate = (category1Rate + category2Rate + category3Rate) / 3;
        // Print the average rate
        System.out.println("Average Rate for all three categories: " + averageRate);
    }
}
iv) public class PrimazonCompany {
    public static void main(String[] args) {
        // Given values
        double category1Value = 12000;
        double category2Value = 10000;
        // The value for category 3 is unknown.
        // Find the minimum value among all three categories
        double minValue = Math.min(category1Value, Math.min(category2Value,
getCategory3Value()));
        // Print the minimum value
        System.out.println("Minimum Value for all three categories: " + minValue);
    }
    // Method to get the value for Category 3 (Electronic Gadgets)
    private static double getCategory3Value() {
        // You should calculate or provide the value for Category 3 here.
        // For now, I'll return 0, but you should replace this with the actual value.
        return 0;
    }
}

```

9)Implement polymorphism and inheritance for an ATM transactions.

**CODE**

```

// Base class Transaction
class Transaction {
    protected double amount;

    public Transaction(double amount) {
        this.amount = amount;
    }

    public void execute() {
        System.out.println("Transaction executed with amount: $" + amount);
    }
}

// Derived class WithdrawalTransaction
class WithdrawalTransaction extends Transaction {
    public WithdrawalTransaction(double amount) {
        super(amount);
    }

    @Override
    public void execute() {
        System.out.println("Withdrawal executed with amount: $" + amount);
    }
}

// Derived class DepositTransaction
class DepositTransaction extends Transaction {
    public DepositTransaction(double amount) {
        super(amount);
    }

    @Override
    public void execute() {
        System.out.println("Deposit executed with amount: $" + amount);
    }
}

```

```

}

public class ATM {

    public static void main(String[] args) {

        Transaction transaction1 = new WithdrawalTransaction(100.0);

        Transaction transaction2 = new DepositTransaction(200.0);

        transaction1.execute(); // Executes a withdrawal

        transaction2.execute(); // Executes a deposit

    }

}

```

10)Anex is an animation company that produces images which is of pixel resolutions such as 12.2,13.7,15.3,17.2,18.1 pixel resolutions. It wants to calculate the mean, median, values of the resolution data.

### **CODE**

```

import java.util.Arrays;

public class PixelResolutionCalculator {

    public static void main(String[] args) {

        // Pixel resolutions

        double[] resolutions = { 12.2, 13.7, 15.3, 17.2, 18.1 };

        // Calculate the mean resolution

        double mean = calculateMean(resolutions);

        // Calculate the median resolution

        double median = calculateMedian(resolutions);

        // Print the results

        System.out.println("Mean Resolution: " + mean);

        System.out.println("Median Resolution: " + median);

    }

    public static double calculateMean(double[] resolutions) {

        double sum = 0;

        for (double resolution : resolutions) {

            sum += resolution;

        }

    }
}

```

```

    }
    return sum / resolutions.length;
}

public static double calculateMedian(double[] resolutions) {
    Arrays.sort(resolutions);
    int middle = resolutions.length / 2;
    if (resolutions.length % 2 == 0) {
        return (resolutions[middle - 1] + resolutions[middle]) / 2.0;
    } else {
        return resolutions[middle];
    }
}
}

```

11) Using a Java program differentiate recursion and looping .

### **CODE**

```

public class RecursionVsLooping {
    // Recursive method to calculate factorial
    public static int factorialRecursive(int n) {
        if (n == 0 || n == 1) {
            return 1;
        }
        return n * factorialRecursive(n - 1);
    }

    // Iterative method to calculate factorial
    public static int factorialLoop(int n) {
        int result = 1;
        for (int i = 1; i <= n; i++) {
            result *= i;
        }
    }
}

```

```

    }
    return result;
}

public static void main(String[] args) {
    int n = 5;
    // Calculate factorial using recursion
    int recursiveResult = factorialRecursive(n);
    System.out.println("Factorial using recursion: " + recursiveResult);

    // Calculate factorial using looping
    int loopResult = factorialLoop(n);
    System.out.println("Factorial using looping: " + loopResult);
}
}

```

12) Using Java to find out 2<sup>nd</sup> maximum element in the array.

### **CODE**

```

public class SecondMaxElement {
    public static void main(String[] args) {
        int[] numbers = {5, 12, 8, 15, 20, 10};

        int firstMax = Integer.MIN_VALUE; // Initialize to smallest possible value
        int secondMax = Integer.MIN_VALUE; // Initialize to smallest possible value

        for (int number : numbers) {
            if (number > firstMax) {
                secondMax = firstMax;
                firstMax = number;
            } else if (number > secondMax && number != firstMax) {

```

```

        secondMax = number;
    }
}

if (secondMax != Integer.MIN_VALUE) {
    System.out.println("Second Maximum Element: " + secondMax);
} else {
    System.out.println("No second maximum element found.");
}
}
}

```

13) Using Java program perform the following operations in a string

- i) Making 1<sup>st</sup> character of string as Capital
- ii) Separating vowels and print the remaining string
- iii) Reversing an alternate character in the string
- iv) Replacing special characters instead of vowels in a string

### CODE

```

i) public class StringOperations {
    public static void main(String[] args) {
        String inputString = "hello, world!";
        // Check if the input string is not empty
        if (!inputString.isEmpty()) {
            // Make the first character uppercase
            String outputString = inputString.substring(0, 1).toUpperCase() +
inputString.substring(1);
            // Print the modified string
            System.out.println("Original string: " + inputString);
            System.out.println("String with the first character capitalized: " + outputString);
        } else {

```

```

        System.out.println("The input string is empty.");
    }
}
}

```

```

ii) public class StringOperations {
    public static void main(String[] args) {
        String inputString = "Hello, World!";
        // Remove vowels from the string
        String remainingString = removeVowels(inputString);
        // Print the remaining string
        System.out.println("Original string: " + inputString);
        System.out.println("Remaining string without vowels: " + remainingString);
    }
    public static String removeVowels(String input) {
        // Regular expression to match vowels (both uppercase and lowercase)
        String pattern = "[aeiouAEIOU]";
        // Remove vowels from the input string and return the result
        return input.replaceAll(pattern, "");
    }
}

```

```

iii) public class StringOperations {
    public static void main(String[] args) {
        String inputString = "Hello, World!";
        // Reverse alternate characters in the string
        String reversedString = reverseAlternateChars(inputString);
        // Print the modified string
        System.out.println("Original string: " + inputString);
        System.out.println("String with alternate characters reversed: " + reversedString);
    }
}

```

```

    }

    public static String reverseAlternateChars(String input) {
        char[] charArray = input.toCharArray();
        // Traverse the string, starting from index 0 (even index)
        for (int i = 0; i < charArray.length; i += 2) {
            // Check if the next character is within the string bounds
            if (i + 1 < charArray.length) {
                char temp = charArray[i];
                charArray[i] = charArray[i + 1];
                charArray[i + 1] = temp;
            }
        }
        // Build the modified string
        return new String(charArray);
    }
}

iv) public class StringOperations {
    public static void main(String[] args) {
        String inputString = "Hello, World!";
        // Replace vowels with special characters in the string
        String replacedString = replaceVowelsWithSpecialChars(inputString);
        // Print the modified string
        System.out.println("Original string: " + inputString);
        System.out.println("String with vowels replaced by special characters: " +
replacedString);
    }

    public static String replaceVowelsWithSpecialChars(String input) {
        StringBuilder result = new StringBuilder();
        for (char ch : input.toCharArray()) {
            // Check if the character is a vowel (uppercase or lowercase)

```



```

        if ("aeiouAEIOU".indexOf(ch) != -1) {
            // Replace vowels with a special character (e.g., *)
            result.append('*');
        } else {
            // Keep non-vowel characters as they are
            result.append(ch);
        }
    }
    return result.toString();
}
}

```

14) Write a Java program to differentiate abstract class and interface

#### **CODE**

```

// Abstract class
abstract class AbstractClass {
    // Abstract method
    public abstract void abstractMethod();

    // Concrete method
    public void concreteMethod() {
        System.out.println("Concrete method in AbstractClass");
    }
}

// Interface
interface MyInterface {
    // Abstract method
    void interfaceMethod();

    // Default method (Java 8+)
    default void defaultMethod() {
        System.out.println("Default method in MyInterface");
    }
}

```

```

    }
}
// Concrete class implementing the interface and extending the abstract class
class ConcreteClass extends AbstractClass implements MyInterface {
    // Implementing abstract method from AbstractClass
    @Override
    public void abstractMethod() {
        System.out.println("Abstract method implementation in ConcreteClass");
    }
    // Implementing abstract method from MyInterface
    @Override
    public void interfaceMethod() {
        System.out.println("Interface method implementation in ConcreteClass");
    }
}
public class Main {
    public static void main(String[] args) {
        // Create an instance of ConcreteClass
        ConcreteClass instance = new ConcreteClass();
        // Call methods
        instance.abstractMethod();
        instance.interfaceMethod();
        instance.concreteMethod();
        // Call default method from the interface
        instance.defaultMethod();
    }
}

```

15) Develop a Java Program to find out the duplicate values in an integer array

**CODE**

```
import java.util.HashSet;

public class FindDuplicates {

    public static void main(String[] args) {

        int[] numbers = {1, 2, 3, 4, 2, 5, 6, 3, 7, 8, 9, 1, 10};

        // Create a HashSet to store unique values
        HashSet<Integer> uniqueValues = new HashSet<>();

        System.out.println("Duplicate values in the array:");

        for (int number : numbers) {

            // If the number is already in the HashSet, it's a duplicate
            if (!uniqueValues.add(number)) {

                System.out.println(number);

            }

        }

    }

}
```