Experiment-1.1

Student Name: Vaibhav Kumar **UID:** 23BCS13386
Branch: B.E-C.S.E **Section/Group:** 23KRG-2B
Semester: 5th **Date of Performance:** 18/08/2025
Subject Name: PBLJ **Subject Code:** 23CSH-304

Easy Level

1. **Aim:** Create Java programs to manage product details, library systems, and student information using classes, inheritance, and abstraction.

2. **Objective:** To understand string manipulation in Java.

3. **Input/Apparatus Used:** Java basic input and string handling.

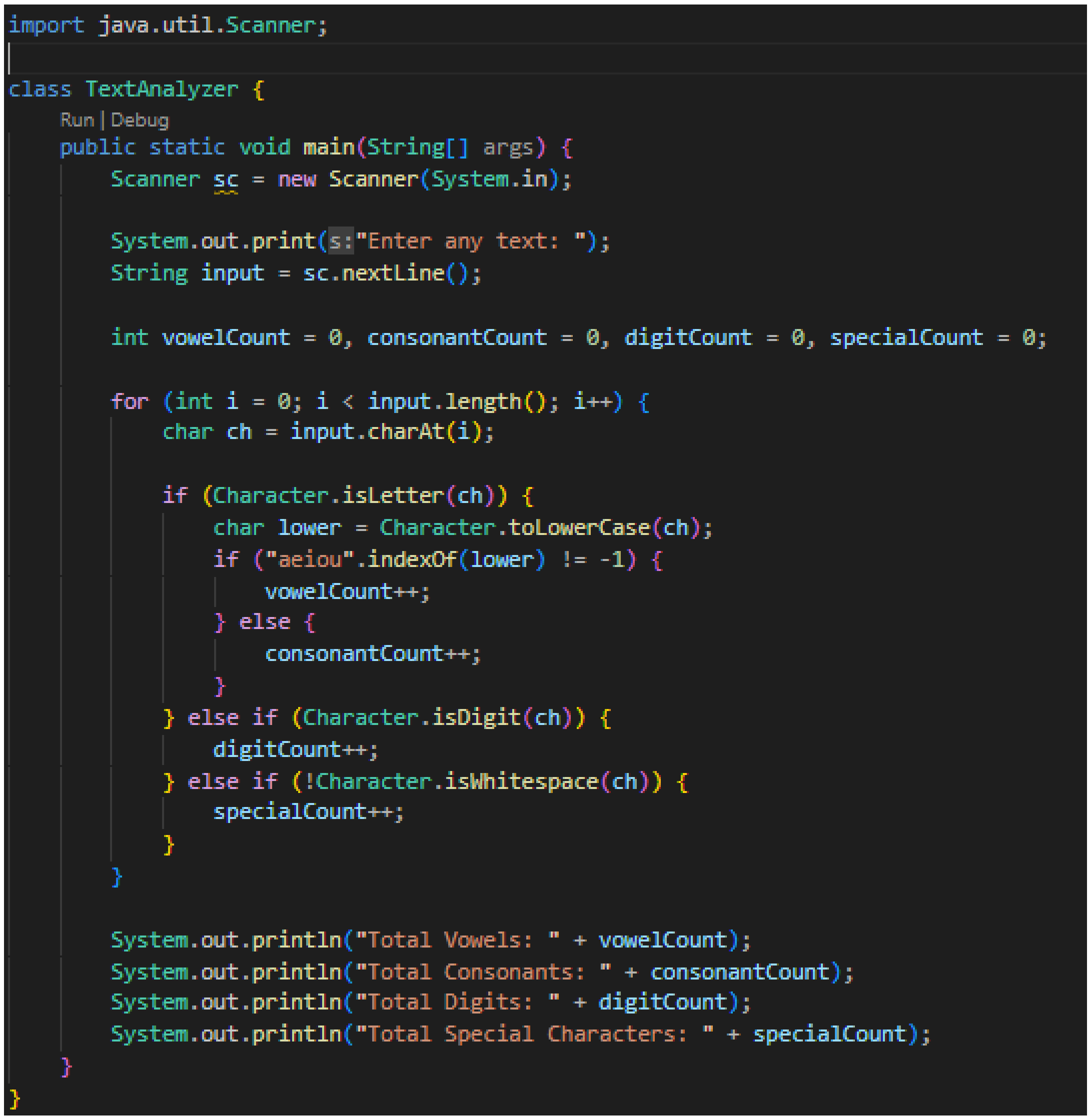
4. **Procedure:**
Step1: Prompt the user to enter a string.
Step2: Traverse each character in the string.
Step3: Classify each character using conditions:
 - If the character is a vowel (a, e, i, o, u), increment the vowel count.
 - If it is a consonant (alphabetic and not a vowel), increment the consonant count.
 - If it is a digit (0–9), increment the digit count.
 - If it is none of the above and not a space, it is a special character.
Step4: Print the counts of vowels, consonants, digits, and special characters.

Sample Input:
Enter a string: **Hello World! 123**

Sample Output:
Vowels: 3
Consonants: 7
Digits: 3

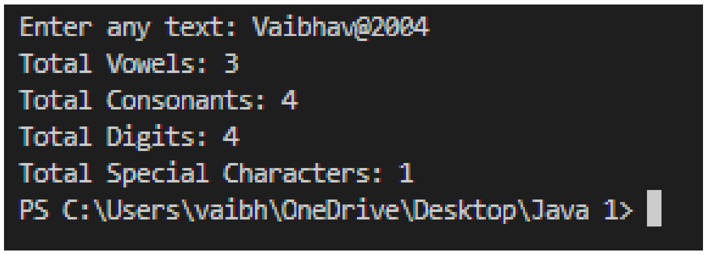
Special Characters: 1

5. Code:



```
import java.util.Scanner;

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

        System.out.print(s:"Enter any text: ");
        String input = sc.nextLine();

        int vowelCount = 0, consonantCount = 0, digitCount = 0, specialCount = 0;

        for (int i = 0; i < input.length(); i++) {
            char ch = input.charAt(i);

            if (Character.isLetter(ch)) {
                char lower = Character.toLowerCase(ch);
                if ("aeiou".indexOf(lower) != -1) {
                    vowelCount++;
                } else {
                    consonantCount++;
                }
            } else if (Character.isDigit(ch)) {
                digitCount++;
            } else if (!Character.isWhitespace(ch)) {
                specialCount++;
            }
        }

        System.out.println("Total Vowels: " + vowelCount);
        System.out.println("Total Consonants: " + consonantCount);
        System.out.println("Total Digits: " + digitCount);
        System.out.println("Total Special Characters: " + specialCount);
    }
}
```

6. Output:



```
Enter any text: Vaibhav@2004
Total Vowels: 3
Total Consonants: 4
Total Digits: 4
Total Special Characters: 1
PS C:\Users\vaibh\OneDrive\Desktop\Java 1>
```

Medium Level

1. **Aim:** Write a Java program to perform matrix operations (addition, subtraction, and multiplication) on two matrices provided by the user. The program should: Check the dimensions of the matrices to ensure valid operations.

2. **Objective:** Understand multidimensional array manipulation and matrix operation validation.

3. **Input/Apparatus Used:** Java multidimensional arrays and control structures.

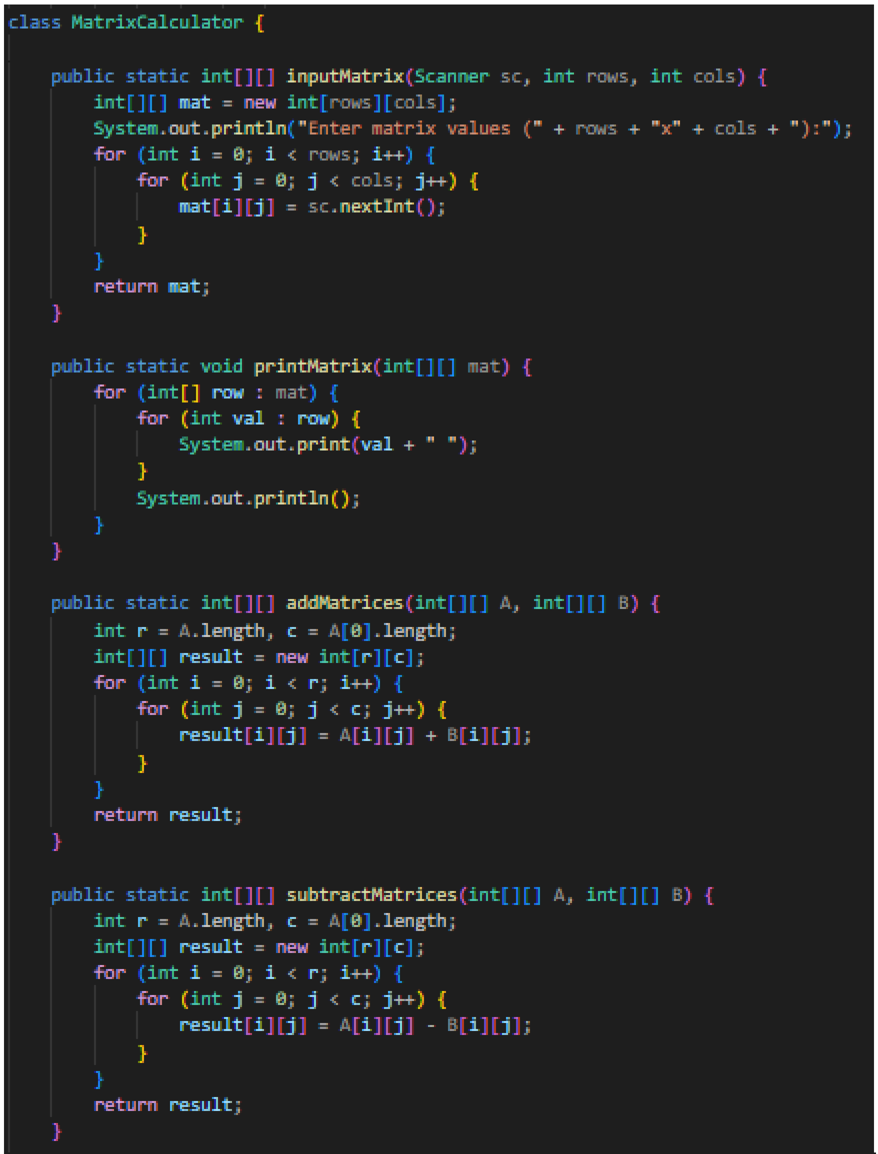
4. **Procedure:**
Step1: Accept input from the user for two matrices (2D arrays).
Step2: Check that the dimensions of matrices are valid for the desired operations:
 - For addition/subtraction: dimensions must be equal.
 - For multiplication: columns of Matrix A = rows of Matrix B.
Step3: Use nested loops to perform:
 - Addition: result[i][j] = matrixA[i][j] + matrixB[i][j]
 - Subtraction: result[i][j] = matrixA[i][j] - matrixB[i][j]
 - Multiplication: result[i][j] = sum(matrixA[i][k] * matrixB[k][j])
Step4: Display the resulting matrices.

Sample Input:
```
Matrix 1:
2 3
4 5

Matrix 2:
6 7
8 9
```

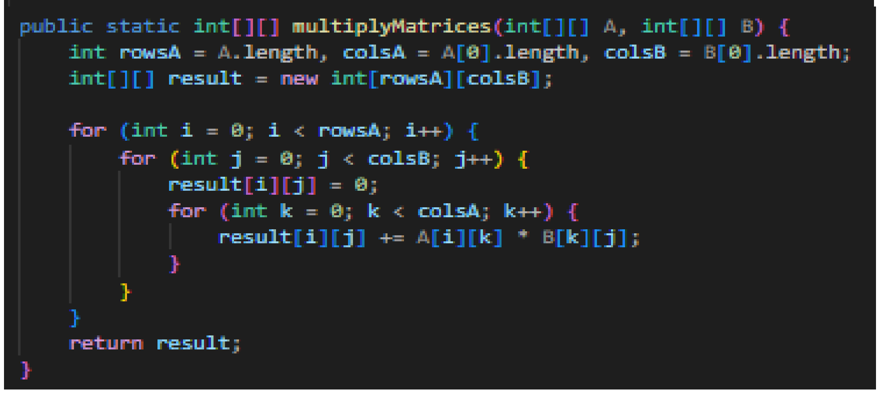
Sample Output:
```
Addition:
8 10
12 14

Subtraction:
-4 -4
-4 -4
```

```
Multiplication:
36 41
64 73
```

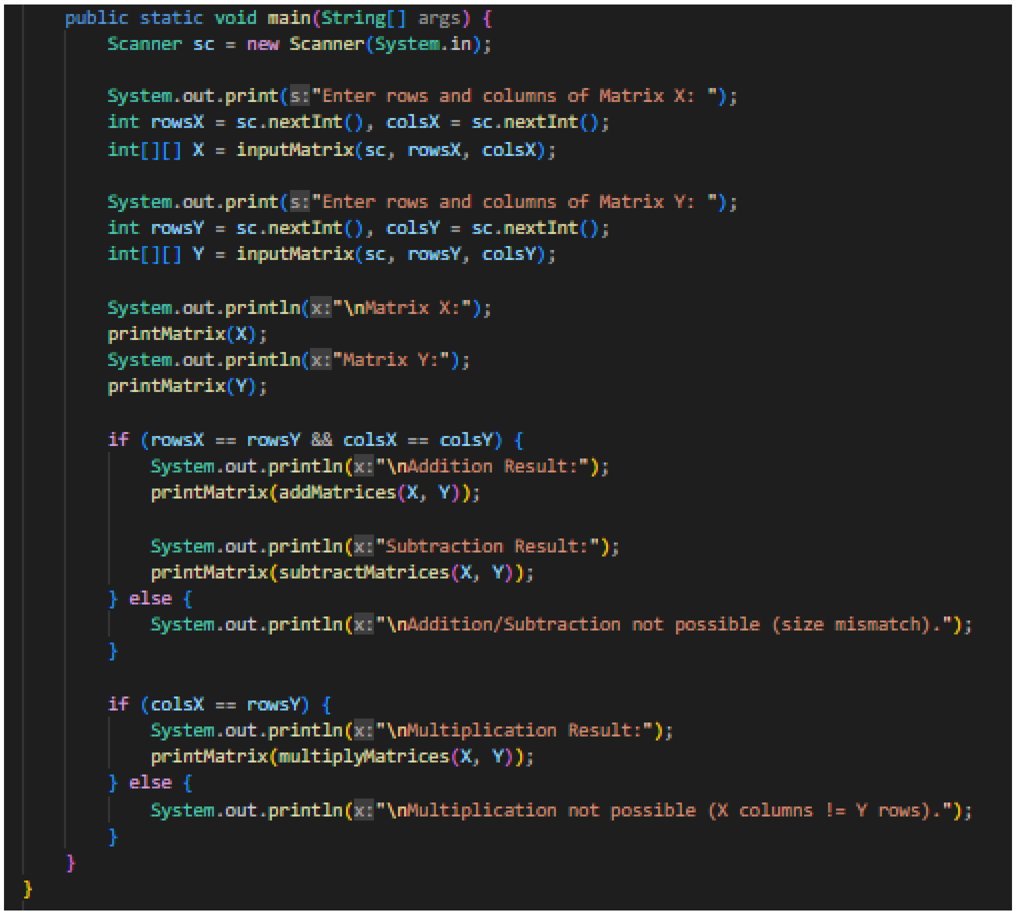
5. Code:



```java
class MatrixCalculator {

    public static int[][] inputMatrix(Scanner sc, int rows, int cols) {
        int[][] mat = new int[rows][cols];
        System.out.println("Enter matrix values (" + rows + "x" + cols + "):");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                mat[i][j] = sc.nextInt();
            }
        }
        return mat;
    }

    public static void printMatrix(int[][] mat) {
        for (int[] row : mat) {
            for (int val : row) {
                System.out.print(val + " ");
            }
            System.out.println();
        }
    }

    public static int[][] addMatrices(int[][] A, int[][] B) {
        int r = A.length, c = A[0].length;
        int[][] result = new int[r][c];
        for (int i = 0; i < r; i++) {
            for (int j = 0; j < c; j++) {
                result[i][j] = A[i][j] + B[i][j];
            }
        }
        return result;
    }

    public static int[][] subtractMatrices(int[][] A, int[][] B) {
        int r = A.length, c = A[0].length;
        int[][] result = new int[r][c];
        for (int i = 0; i < r; i++) {
            for (int j = 0; j < c; j++) {
                result[i][j] = A[i][j] - B[i][j];
            }
        }
        return result;
    }
```

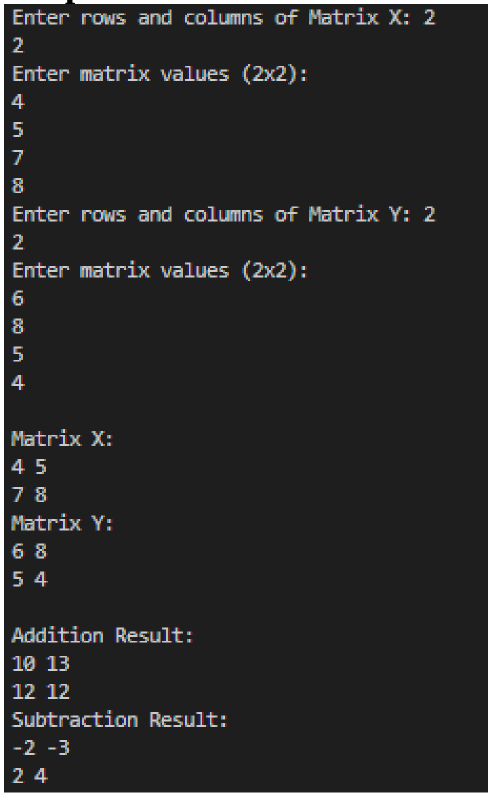


```java
public static int[][] multiplyMatrices(int[][] A, int[][] B) {
    int rowsA = A.length, colsA = A[0].length, colsB = B[0].length;
    int[][] result = new int[rowsA][colsB];

    for (int i = 0; i < rowsA; i++) {
        for (int j = 0; j < colsB; j++) {
            result[i][j] = 0;
            for (int k = 0; k < colsA; k++) {
                result[i][j] += A[i][k] * B[k][j];
            }
        }
    }
    return result;
}
```

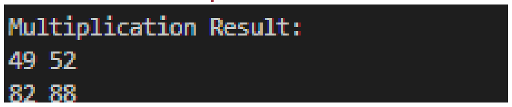


```
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print(s:"Enter rows and columns of Matrix X: ");
        int rowsX = sc.nextInt(), colsX = sc.nextInt();
        int[][] X = inputMatrix(sc, rowsX, colsX);

        System.out.print(s:"Enter rows and columns of Matrix Y: ");
        int rowsY = sc.nextInt(), colsY = sc.nextInt();
        int[][] Y = inputMatrix(sc, rowsY, colsY);

        System.out.println(x:"\nMatrix X:");
        printMatrix(X);
        System.out.println(x:"Matrix Y:");
        printMatrix(Y);

        if (rowsX == rowsY && colsX == colsY) {
            System.out.println(x:"\nAddition Result:");
            printMatrix(addMatrices(X, Y));

            System.out.println(x:"Subtraction Result:");
            printMatrix(subtractMatrices(X, Y));
        } else {
            System.out.println(x:"\nAddition/Subtraction not possible (size mismatch).");
        }

        if (colsX == rowsY) {
            System.out.println(x:"\nMultiplication Result:");
            printMatrix(multiplyMatrices(X, Y));
        } else {
            System.out.println(x:"\nMultiplication not possible (X columns != Y rows).");
        }
    }
}
```

6. Output:



```
Enter rows and columns of Matrix X: 2
2
Enter matrix values (2x2):
4
5
7
8
Enter rows and columns of Matrix Y: 2
2
Enter matrix values (2x2):
6
8
5
4

Matrix X:
4 5
7 8
Matrix Y:
6 8
5 4

Addition Result:
10 13
12 12
Subtraction Result:
-2 -3
2 4
```



```
Multiplication Result:
49 52
82 88
```

Hard Level

1. **Aim:** Create a Java program to implement a basic banking system with the following features: Account creation (Name, Account Number). Deposit and withdrawal operations. Prevent overdraft by checking the balance before withdrawal.

2. **Objective:** Apply object-oriented programming concepts in a practical system.

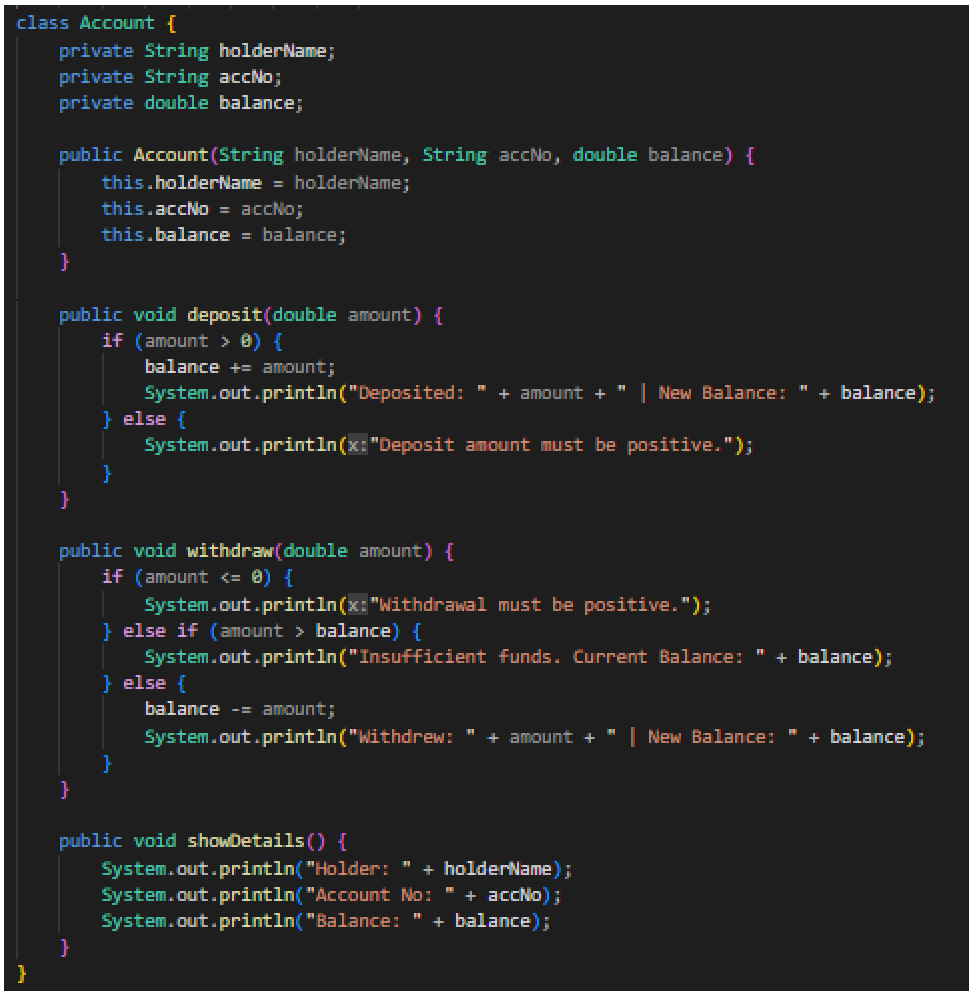
3. **Input/Apparatus Used:** Java classes, objects, and control structures.

4. **Procedure:**
Step1: Define a `BankAccount` class with fields like name, account number, and balance.
Step2: Implement methods for:
   - deposit(double amount): Adds amount to balance.
   - withdraw(double amount): Checks balance before subtracting.
Step3: In the main program, create a new account by taking user input.
Step4: Allow the user to perform deposit and withdrawal operations.
Step5: Display appropriate messages and updated balances.

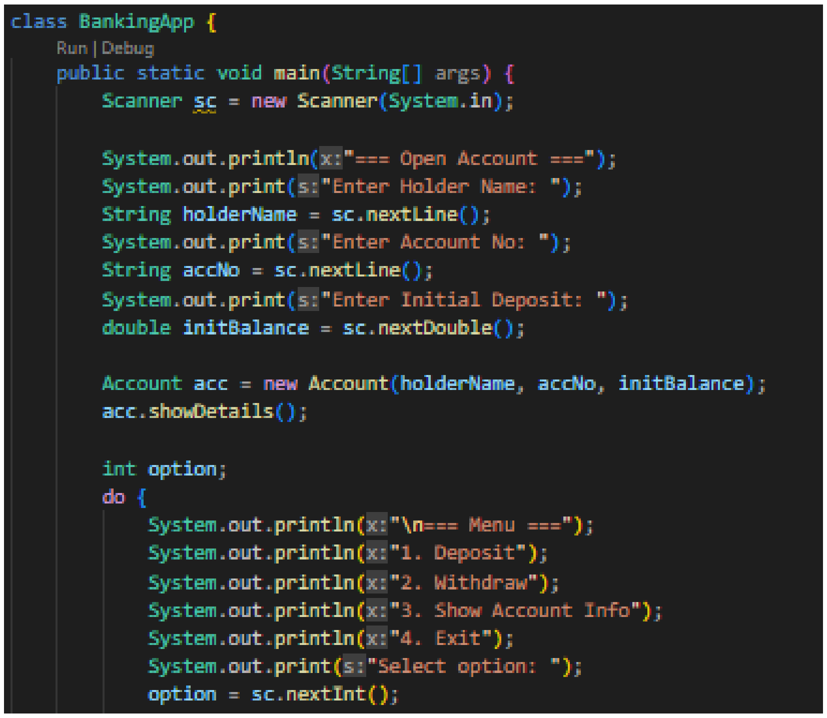
Sample Input:
Create Account:
Name: Vaibhav Kumar
Account Number: 12456
Initial Balance: ₹5000
Deposit: ₹2000
Withdraw: ₹6000

Sample Output:
Deposit successful! Current Balance: ₹7000
Error: Insufficient funds. Current Balance:₹7000

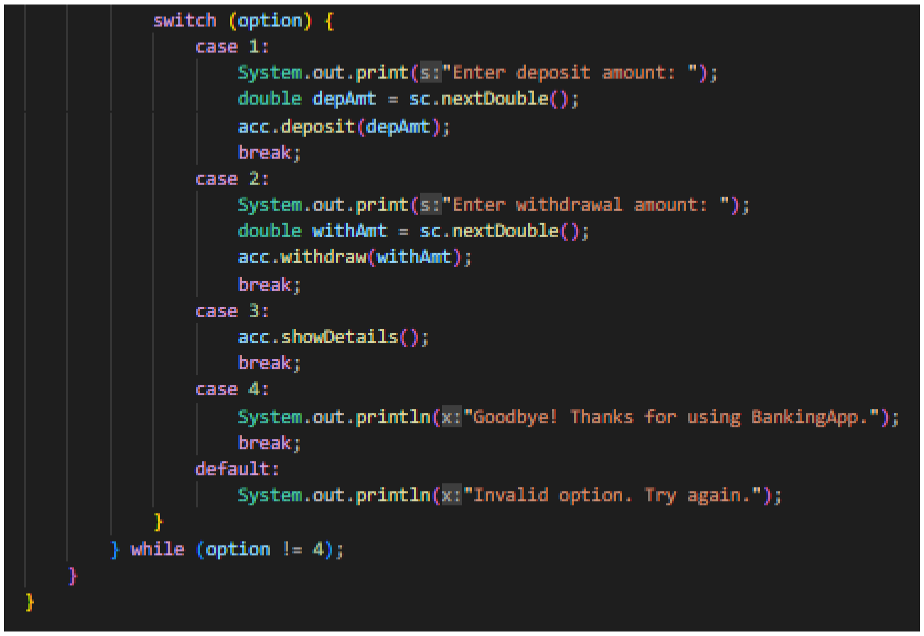
5. **Code:**



```java
class Account {
    private String holderName;
    private String accNo;
    private double balance;

    public Account(String holderName, String accNo, double balance) {
        this.holderName = holderName;
        this.accNo = accNo;
        this.balance = balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount + " | New Balance: " + balance);
        } else {
            System.out.println(x:"Deposit amount must be positive.");
        }
    }

    public void withdraw(double amount) {
        if (amount <= 0) {
            System.out.println(x:"Withdrawal must be positive.");
        } else if (amount > balance) {
            System.out.println("Insufficient funds. Current Balance: " + balance);
        } else {
            balance -= amount;
            System.out.println("Withdrew: " + amount + " | New Balance: " + balance);
        }
    }

    public void showDetails() {
        System.out.println("Holder: " + holderName);
        System.out.println("Account No: " + accNo);
        System.out.println("Balance: " + balance);
    }
}
```

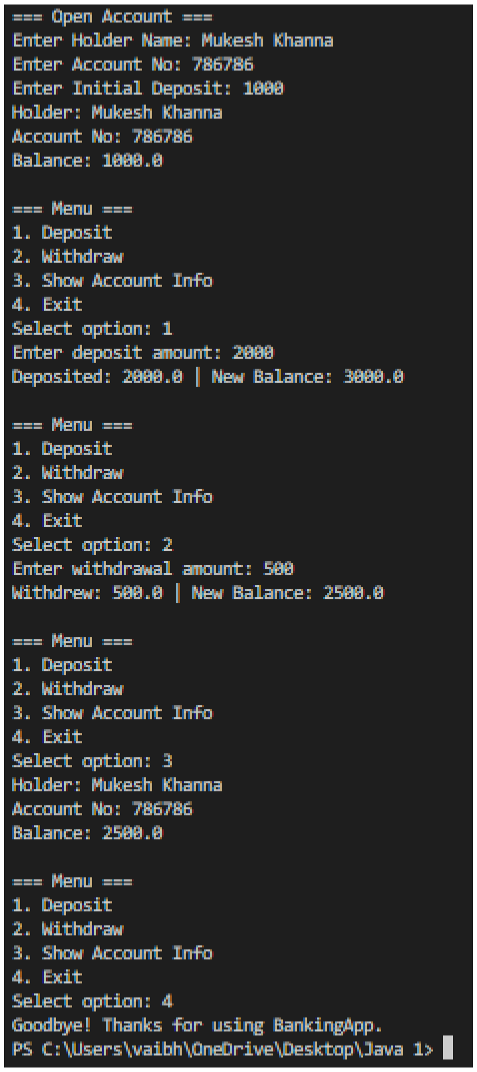


```java
class BankingApp {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println(x:"=== Open Account ===");
        System.out.print(s:"Enter Holder Name: ");
        String holderName = sc.nextLine();
        System.out.print(s:"Enter Account No: ");
        String accNo = sc.nextLine();
        System.out.print(s:"Enter Initial Deposit: ");
        double initBalance = sc.nextDouble();

        Account acc = new Account(holderName, accNo, initBalance);
        acc.showDetails();

        int option;
        do {
            System.out.println(x:"\n=== Menu ===");
            System.out.println(x:"1. Deposit");
            System.out.println(x:"2. Withdraw");
            System.out.println(x:"3. Show Account Info");
            System.out.println(x:"4. Exit");
            System.out.print(s:"Select option: ");
            option = sc.nextInt();
```



```
                switch (option) {
                    case 1:
                        System.out.print(s:"Enter deposit amount: ");
                        double depAmt = sc.nextDouble();
                        acc.deposit(depAmt);
                        break;
                    case 2:
                        System.out.print(s:"Enter withdrawal amount: ");
                        double withAmt = sc.nextDouble();
                        acc.withdraw(withAmt);
                        break;
                    case 3:
                        acc.showDetails();
                        break;
                    case 4:
                        System.out.println(x:"Goodbye! Thanks for using BankingApp.");
                        break;
                    default:
                        System.out.println(x:"Invalid option. Try again.");
                }
            } while (option != 4);
        }
    }
```

6. Output:



```
=== Open Account ===
Enter Holder Name: Mukesh Khanna
Enter Account No: 786786
Enter Initial Deposit: 1000
Holder: Mukesh Khanna
Account No: 786786
Balance: 1000.0

=== Menu ===
1. Deposit
2. Withdraw
3. Show Account Info
4. Exit
Select option: 1
Enter deposit amount: 2000
Deposited: 2000.0 | New Balance: 3000.0

=== Menu ===
1. Deposit
2. Withdraw
3. Show Account Info
4. Exit
Select option: 2
Enter withdrawal amount: 500
Withdrew: 500.0 | New Balance: 2500.0

=== Menu ===
1. Deposit
2. Withdraw
3. Show Account Info
4. Exit
Select option: 3
Holder: Mukesh Khanna
Account No: 786786
Balance: 2500.0

=== Menu ===
1. Deposit
2. Withdraw
3. Show Account Info
4. Exit
Select option: 4
Goodbye! Thanks for using BankingApp.
PS C:\Users\vaibh\OneDrive\Desktop\Java 1>
```