



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment - 1

Student Name: Yash Goel

Branch: BE-CSE

Semester: 5th

Subject Name: Project Based Learning in Java

Subject Code: 23CSH-304

UID: 23BCS11498

Section/Group: KRG-2B

Date of Performance: 12/8/25

Aim: To develop Java programs to analyze strings, perform matrix operations, and implement basic banking system functionality.

Easy-level Problem-

Aim: To write a Java program to analyze a string input by the user. The program should: Count the number of vowels, consonants, digits and special characters in the string.

Objective: To understand string manipulation in Java using concepts like java basic input and string handling.

Procedure:

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

Sample Input -

Enter a string: Hello World 2025!

Sample Output -

Vowels : 3

Consonants : 7

Digits: 4

Special characters: 1



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Code –

```
import java.util.Scanner;

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

        System.out.print("Enter a string: ");
        String str = sc.nextLine();

        int vowels = 0, consonants = 0, digits = 0, special = 0;

        str = str.toLowerCase();

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

            if (ch >= 'a' && ch <= 'z') {
                if ("aeiou".indexOf(ch) != -1) {
                    vowels++;
                } else {
                    consonants++;
                }
            } else if (ch >= '0' && ch <= '9') {
                digits++;
            } else if (ch != ' ') {
                special++;
            }
        }

        System.out.println("Vowels: " + vowels);
        System.out.println("Consonants: " + consonants);
        System.out.println("Digits: " + digits);
        System.out.println("Special Characters: " + special);

        sc.close();
    }
}
```

Output -

```
PS D:\3rd_year\java> cd "d:\3rd_year\java\" ; if ($?) { javac easy.java } ; if ($?) { java easy }
Enter a string: Hello World 2025!
Vowels: 3
Consonants: 7
Digits: 4
Special Characters: 1
PS D:\3rd_year\java> █
```

Medium- Level Problem -

Aim : To write a Java program to perform matrix operations(addition, subtraction, and multiplication) on two matrices provided by the user. The program need to check the dimensions of the matrices to ensure valid operations.

Objective: Understand multidimensional array manipulation and matrix operation validation using concepts of Java multidimensional array and control structures.

Procedure:

1. Accept user input for 2 matrices (2D arrays).
2. 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.
3. Use nested loops to perform:
 - Addition : $\text{result}[i][j] = \text{matrixA}[i][j] + \text{matrixB}[i][j]$
 - Subtraction : $\text{result}[i][j] = \text{matrixA}[i][j] - \text{matrixB}[i][j]$
 - Multiplication : $\text{result}[i][j] = \text{sum}(\text{matrixA}[i][k] * \text{matrixB}[k][j])$
4. Display the resulting matrices.

Sample Input :

Matrix 1: 1 2
 3 4
Matrix 2: 5 6
 7 8

Sample Output:

Addition:
6 8
10 12
Subtraction:
-4 -4
-4 -4

Multiplication:19 22
 43 50



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Code :

```
import java.util.Scanner;

public class medium {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter rows and columns of first matrix: ");
        int r1 = sc.nextInt();
        int c1 = sc.nextInt();
        System.out.print("Enter rows and columns of second matrix: ");
        int r2 = sc.nextInt();
        int c2 = sc.nextInt();
        int[][] A = new int[r1][c1];
        int[][] B = new int[r2][c2];
        System.out.println("Enter elements of first matrix:");
        for (int i = 0; i < r1; i++) {
            for (int j = 0; j < c1; j++) {
                A[i][j] = sc.nextInt();
            }
        }
        System.out.println("Enter elements of second matrix:");
        for (int i = 0; i < r2; i++) {
            for (int j = 0; j < c2; j++) {
                B[i][j] = sc.nextInt();
            }
        }
        if (r1 == r2 && c1 == c2) {
            System.out.println("Addition:");
            for (int i = 0; i < r1; i++) {
                for (int j = 0; j < c1; j++) {
                    System.out.print((A[i][j] + B[i][j]) + " ");
                }
                System.out.println();
            }
            System.out.println("Subtraction:");
            for (int i = 0; i < r1; i++) {
                for (int j = 0; j < c1; j++) {
                    System.out.print((A[i][j] - B[i][j]) + " ");
                }
            }
        }
    }
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        System.out.println();
    }
} else {
    System.out.println("Addition/Subtraction not possible (dimension mismatch).");
}
if (c1 == r2) {
    System.out.println("Multiplication:");
    int[][] result = new int[r1][c2];
    for (int i = 0; i < r1; i++) {
        for (int j = 0; j < c2; j++) {
            for (int k = 0; k < c1; k++) {
                result[i][j] += A[i][k] * B[k][j];
            }
            System.out.print(result[i][j] + " ");
        }
        System.out.println();
    }
} else {
    System.out.println("Multiplication not possible (dimension mismatch).");
}
sc.close();
}
```

Output:

```
Enter rows and columns of first matrix: 2 2
Enter rows and columns of second matrix: 2 2
Enter elements of first matrix:
1 2
3 4
Enter elements of second matrix:
5 6
7 8
Addition:
6 8
10 12
Subtraction:
-4 -4
-4 -4
Multiplication:
19 22
43 50
PS D:\3rd_year\java> |
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Hard -level Problem-

Aim : To 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.

Objective: Apply object-oriented programming concepts in a practical system using concepts like Java classes, objects and control structures.

Procedure:

1. Define a 'BankAccount' class with fields like name,account number , and balance.
2. Implement methods for:
 - deposit(double amount): Adds amount to balance.
 - withdraw(double amount): checks balance before subtracting.
3. In the main program, create a new account by taking user input.
4. Allow the user to perform deposit and withdrawal operations.
5. Display appropriate messages and updated balances.

Sample Input:

Create Account:

Name: John Doe

Account Number: 12345

Initial Balance: 1000

Deposit: 500

Withdraw: 2000

Sample Output:

Deposit successful! Current Balance: 1500

Error: Insufficient funds. Current Balance: 1500



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Code :

```
import java.util.Scanner;

class BankAccount {
    String name;
    int accountNumber;
    double balance;

    BankAccount(String name, int accountNumber, double initialBalance) {
        this.name = name;
        this.accountNumber = accountNumber;
        this.balance = initialBalance;
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful! Current Balance: " + balance);
    }

    void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Error: Insufficient funds. Current Balance: " +
balance);
        } else {
            balance -= amount;
            System.out.println("Withdrawal successful! Current Balance: " + balance);
        }
    }
}

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

        // Account creation
        System.out.print("Enter Name: ");
        String name = sc.nextLine();

        System.out.print("Enter Account Number: ");
        int accNo = sc.nextInt();

        System.out.print("Enter Initial Balance: ");
        double balance = sc.nextDouble();

        BankAccount account = new BankAccount(name, accNo, balance);

        while (true) {
            System.out.println("\n1. Deposit\n2. Withdraw\n3. Exit");
            System.out.print("Choose an option: ");
            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter deposit amount: ");
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        double dep = sc.nextDouble();
        account.deposit(dep);
        break;

    case 2:
        System.out.print("Enter withdrawal amount: ");
        double wd = sc.nextDouble();
        account.withdraw(wd);
        break;

    case 3:
        System.out.println("Exiting... Thank you!");
        sc.close();
        return;

    default:
        System.out.println("Invalid choice.");
    }
}
}
```

Output:

```
cd "d:\3rd_year\java\" ; if ($?) { javac multiple_inheritance.java } ; if ($?) { java multiple_inheritance }
Enter Account Number: 12345
Enter Initial Balance: 1000

1. Deposit
2. Withdraw
3. Exit
Choose an option: 1
Enter deposit amount: 500
Deposit successful! Current Balance: 1500.0

1. Deposit
2. Withdraw
3. Exit
Choose an option: 2
Enter withdrawal amount: 2000
Error: Insufficient funds. Current Balance: 1500.0

1. Deposit
2. Withdraw
3. Exit
Choose an option: 
```