**FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERIG**

**Department of Computer Engineering**

**Experiment 4- Based on Arrays**

1. **Course Details:**

| **Academic Year** | **2023 - 24** | **Estimated Time** | **Experiment No. 4– 02 Hours** |
| --- | --- | --- | --- |
| **Course & Semester** | **S.E. (COMP) – Sem. III** | **Subject Name** | **Skill based lab Course-OOP with Java** |
| **Module No.** | **03** | **Chapter Title** | **Array, String and Vector** |
| **Experiment Type** | **Software Performance** | **Subject Code** | **CSL304** |
|  |  |  |  |

| **Name of Student** | Vivian Vijay Ludrick | **Roll No.** | 9914 |
| --- | --- | --- | --- |
| **Date of**  **Performance:** | 18/8/23 | **Date of Submission:** | 25/08/23 |
| **CO Mapping** | **CSL304.3: Apply the concept of strings, arrays, and vectors** | | |

| **Timeline**  **(2)** | **Preparedness**  **(2)** | **Effort**  **(3)** | **Result**  **(3)** | **Total (10)** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

**Problem Statements:**

**1.Based on 1-D Array**

a) Write a program to reverse a one-dimensional array. (Using **for each** loop)

**Code:**

public class ReverseArray {

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

int[] arr = {1, 2, 3, 4, 5};

*// prints the original array*

System.out.print("Original Array: ");

for (int num : arr) {

System.out.print(num + " ");

}

*// defined an array having the same length as the original*

int[] reversedArr = new int[arr.length];

int index = arr.length - 1;

*// for each loop to reverse an array*

for (int og : arr) {

reversedArr[index] = og;

index--;

}

*// for loop to print the reversed array*

System.out.print("\nReversed Array: ");

for (int num : reversedArr) {

System.out.print(num + " ");

}

}

}

**OUTPUT:**

****

b) Write a program to Sort a one-dimensional array. (Using **Arrays class** of java and method)

**CODE:**

import java.util.Arrays;

public class SortArray {

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

int[] arr = {5, 2, 8, 1, 9};

System.out.print("Original Array: ");

for (int num : arr) {

System.out.print(num + " ");

}

Arrays.sort(arr);

System.out.print("\nSorted Array: ");

for (int num : arr) {

System.out.print(num + " ");

}

}

}

**OUTPUT:**

****

c) Write a program to calculate Mode, Median and Mean of a one-dimensional array.

**CODE:**

import java.util.Arrays;

public class ArrayStats {

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

int[] arr = { 5, 2, 8, 1, 9, 2, 5, 3, 7 };

System.out.print("Array: ");

for (int num : arr) {

System.out.print(num + " ");

}

*// Calculate mean*

double sum = 0;

for (int num : arr) {

sum += num;

}

double mean = sum / arr.length;

*// Calculate median*

Arrays.sort(arr);

double median;

if (arr.length % 2 == 0) {

median = (arr[arr.length / 2] + arr[arr.length / 2 - 1]) / 2.0;

} else {

median = arr[arr.length / 2];

}

*// Calculate mode*

int mode = arr[0];

int maxCount = 0;

for (int i = 0; i < arr.length; i++) {

int count = 0;

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

if (arr[j] == arr[i]) {

count++;

}

}

if (count > maxCount) {

maxCount = count;

mode = arr[i];

}

}

System.out.println("\nMean: " + mean);

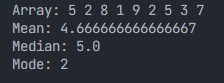
System.out.println("Median: " + median);

System.out.println("Mode: " + mode);

}

}

**OUTPUT:**

****

**2. Based on 2-D Array**

a) Write a program to sum of all elements in an array.

**CODE:**

public class ArraySum {

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

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

System.out.println("Array:");

for (int[] row : arr) {

for (int num : row) {

System.out.print(num + " ");

}

System.out.println();

}

int sum = 0;

for (int[] row : arr) {

for (int num : row) {

sum += num;

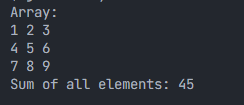
}

}

System.out.println("Sum of all elements: " + sum);

}

}

**OUTPUT:  
**

b) Write a program to sum of diagonal element in an array.

**CODE:**

public class DiagonalSum {

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

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

System.out.println("Array:");

for (int[] row : arr) {

for (int num : row) {

System.out.print(num + " ");

}

System.out.println();

}

*// no need to create a nested loop as the indices will be the same*

int sum = 0;

for (int i = 0; i < arr.length; i++) {

sum += arr[i][i];

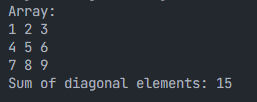
}

System.out.println("Sum of diagonal elements: " + sum);

}

}

**OUTPUT:**

****

c) Write a program to sum of each column in an array

**CODE:**

public class ColumnSum {

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

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

System.out.println("Array:");

for (int[] row : arr) {

for (int num : row) {

System.out.print(num + " ");

}

System.out.println();

}

for (int i = 0; i < arr.length; i++) {

int columnSums = 0;

for (int j = 0; j < arr[i].length; j++) {

columnSums += arr[j][i];

}

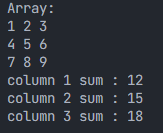
System.out.println("column " + (i + 1) + " sum : " + columnSums);

}

}

}

**OUTPUT:**

****

d) Write a program to find minimum of each row in an array.

**CODE:**

public class RowMin {

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

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

System.out.println("Array:");

for (int[] row : arr) {

for (int num : row) {

System.out.print(num + " ");

}

System.out.println();

}

for (int i = 0; i < arr.length; i++) {

int min = arr[i][0];

for (int j = 1; j < arr[i].length; j++) {

if (arr[i][j] < min) {

min = arr[i][j];

}

}

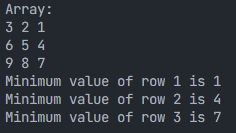
System.out.println("Minimum value of row " + (i + 1) + " is " + min);

}

}

}

**OUTPUT:**

****

e) Write a program to find maximum of each column and its index in a new array.

**CODE:**

public class ColumnMax {

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

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

System.out.println("Array:");

for (int[] row : arr) {

for (int num : row) {

System.out.print(num + " ");

}

System.out.println();

}

int[] columnMaxValues = new int[arr[0].length];

int[] columnMaxIndices = new int[arr[0].length];

for (int j = 0; j < arr[0].length; j++) {

int max = arr[0][j];

int maxIndex = 0;

for (int i = 1; i < arr.length; i++) {

if (arr[i][j] > max) {

max = arr[i][j];

maxIndex = i;

}

}

columnMaxValues[j] = max;

columnMaxIndices[j] = maxIndex;

}

System.out.println("Maximum value of each column:");

for (int i = 0; i < columnMaxValues.length; i++) {

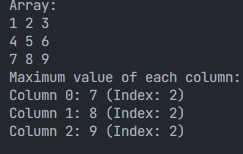
System.out.println("Column " + i + ": " + columnMaxValues[i] + " (Index: " + columnMaxIndices[i] + ")");

}

}

}

**OUTPUT:**

****

f) Write a program to multiply two Arrays.

**CODE:**

public class ArrayMultiplication {

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

int[][] arr1 = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

int[][] arr2 = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};

System.out.println("Array 1:");

for (int[] row : arr1) {

for (int num : row) {

System.out.print(num + " ");

}

System.out.println();

}

System.out.println("Array 2:");

for (int[] row : arr2) {

for (int num : row) {

System.out.print(num + " ");

}

System.out.println();

}

int[][] product = new int[arr1.length][arr1[0].length];

for (int i = 0; i < arr1.length; i++) {

for (int j = 0; j < arr1[i].length; j++) {

product[i][j] = arr1[i][j] \* arr2[i][j];

}

}

System.out.println("Product of arrays:");

for (int[] row : product) {

for (int num : row) {

System.out.print(num + " ");

}

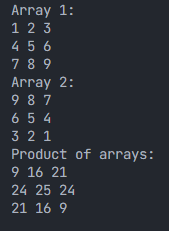
System.out.println();

}

}

}

**OUTPUT:**

****