



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

Experiment No. 6

To use 2D arrays and Strings for solving given problem.

Date of Performance:

Date of Submission:

**Aim:** To use 2D arrays and Strings for solving given problem.

**Objective:** To use 2D array concept and strings in java to solve real world problem

### Theory:

- An array is used to store a fixed-size sequential collection of data of the same type.
- An array can be init in two ways:
  1. Initializing at the time of declaration:  
`dataType[] myArray = {value0, value1, ..., valuek};`
  2. Dynamic declaration:  
`dataType[] myArray = new dataType[arraySize];`  
`myArray[index] = value;`
- Two – dimensional array is the simplest form of a multidimensional array. Data of only same data type can be stored in a 2D array. Data in a 2D Array is stored in a tabular manner which can be represented as a matrix.
- A 2D Array can be declared in 2 ways:
  1. Initializing at the time of declaration:  
`dataType[][] myArray = { {valueR1C1, valueR1C2...}, {valueR2C1, valueR2C2...},...}`
  2. Dynamic declaration:  
`dataType[][] myArray = new dataType[x][y];`  
`myArray[row_index][column_index] = value;`

In Java, string is basically an object that represents sequence of char values. An array of characters works same as Java string. **Java String** class provides a lot of methods to perform operations on strings



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

such as `compare()`, `concat()`, `equals()`, `split()`, `length()`, `replace()`, `compareTo()`, `intern()`, `substring()` etc.

### 1.String literal

To make Java more memory efficient (because no new objects are created if it exists already in the string constant pool).

#### **Example:**

```
String demoString = "GeeksforGeeks";
```

### 2. Using new keyword

- `String s = new String("Welcome");`
- In such a case, JVM will create a new string object in normal (non-pool) heap memory and the literal "Welcome" will be placed in the string constant pool. The variable `s` will refer to the object in the heap (non-pool)

#### **Example:**

```
String demoString = new String ("GeeksforGeeks");
```



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

### Code:

```
import java.util.Scanner;

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

        System.out.print("Enter the number of students: ");
        int numStudents = scanner.nextInt();
        scanner.nextLine();

        String[][] students = new String[numStudents][2];

        for (int i = 0; i < numStudents; i++) {
            System.out.print("Enter name for student " + (i + 1) + ": ");
            students[i][0] = scanner.nextLine();
            System.out.print("Enter score for student " + (i + 1) + ": ");
            students[i][1] = scanner.nextLine();
        }

        System.out.println("\nStudents and their scores:");
        for (int i = 0; i < students.length; i++) {
            System.out.println("Student: " + students[i][0] + ", Score: " + students[i][1]);
        }

        double averageScore = calculateAverageScore(students);
        System.out.println("\nAverage Score: " + averageScore);

        scanner.close();
    }
}
```



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

```
public static double calculateAverageScore(String[][] students) {  
    int totalScore = 0;  
    for (int i = 0; i < students.length; i++) {  
        totalScore += Integer.parseInt(students[i][1]);  
    }  
    return (float) totalScore / students.length;  
}
```

### Output:

Enter the number of students: 3

Enter name for student 1: Jacob

Enter score for student 1: 60

Enter name for student 2: Sam

Enter score for student 2: 48

Enter name for student 3: Haby

Enter score for student 3: 52

Students and their scores:

Student: Jacob, Score: 60

Student: Sam, Score: 48

Student: Haby, Score: 52

Average Score: 53.333333333333336



# Vidyavardhini's College of Engineering and Technology

## Department of Artificial Intelligence & Data Science

### Screenshot:

```
dd231181a0b3bb1e5834c\redhat.java\jdk_ws\Java Programs_fde9a6b3\bin' 'StudentScores'

Enter the number of students: 3

Enter name for student 1: Jacob
Enter score for student 1: 60

Enter name for student 2: Sam
Enter score for student 2: 48

Enter name for student 3: Haby
Enter score for student 3: 52

Students and their scores:
Student: Jacob, Score: 60
Student: Sam, Score: 48
Student: Haby, Score: 52

Average Score: 53.33333333333336
PS E:\Java Programs>

Enter name for student 1: Jacob
Enter score for student 1: 60

Enter name for student 2: Sam
Enter score for student 2: 48

Enter name for student 3: Haby
Enter score for student 3: 52

Students and their scores:
Student: Jacob, Score: 60
Enter score for student 3: 52

Students and their scores:
Student: Jacob, Score: 60
Student: Sam, Score: 48
Student: Haby, Score: 52
Student: Jacob, Score: 60
Student: Sam, Score: 48
Student: Haby, Score: 52
Student: Sam, Score: 48
Student: Haby, Score: 52
Student: Sam, Score: 48
Student: Haby, Score: 52

Average Score: 53.33333333333336
```

### Conclusion:

ANS: Comment on how you have used the concept of string and 2D array.

In this program, a 2D array is used to store student names and their respective scores, where each row represents a student, and the two columns store the student's name and score. The array is dynamically initialized based on user input for the number of students. The names and scores are handled as strings, with the scores being stored as strings initially and later converted to integers for calculations. String operations such as reading input via `scanner.nextLine()` and `Integer.parseInt()` for conversion are utilized. This demonstrates efficient storage and manipulation of string data in a 2D array structure.