|  |
| --- |
| Experiment No. 6 |
| Implement a program on 2D array & strings functions. |
| 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};

1. 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. Intializing at the time of declaration:

dataType[][] myArray = { {valueR1C1, valueR1C2...}, {valueR2C1, valueR2C2...},..}

1. Dynamic declaration:

**dataType[][] myArray = new dataType[x][y];**

**myArray[row\_index][column\_index] = value;**

In [Java](https://www.javatpoint.com/java-tutorial), string is basically an object that represents sequence of char values. An [array](https://www.javatpoint.com/array-in-java) of characters works same as Java string. **Java String** class provides a lot of methods to perform operations on strings 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”);

**Code:**

**2D array Example**

public class TwoDimensionalArrayExample {

public static void main(String[] args) {

// Declare and initialize a 2D array

int[][] twoDArray = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

// Display the elements of the 2D array

System.out.println("Elements of the 2D array:");

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

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

System.out.print(twoDArray[i][j] + " ");

}

System.out.println(); // Move to the next row

}

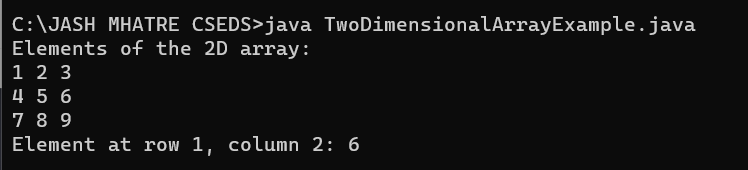
// Access a specific element

int element = twoDArray[1][2];

System.out.println("Element at row 1, column 2: " + element);

}

}

****

**STRING FUNCTION EXAMPLE**

**Code:**

public class Stringoperation

{

public static void main(String args[])

{

String s="JASH MHATRE";

System.out.println(s.toUpperCase());

System.out.println(s.toLowerCase());

System.out.println(s.trim());

System.out.println(s.startsWith("V"));

System.out.println(s.endsWith("a"));

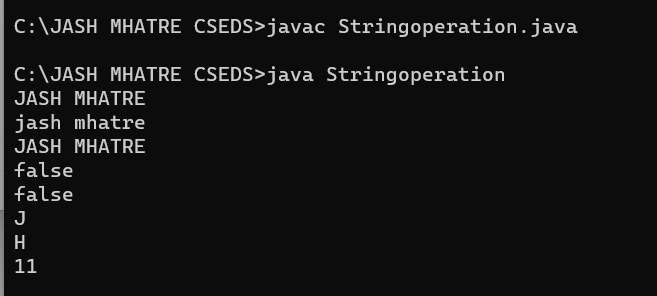
System.out.println(s.charAt(0));

System.out.println(s.charAt(3));

System.out.println(s.length());

}

}

****

**Conclusion:**

2D arrays are a versatile and essential data structure in programming, mathematics, and various scientific fields. They provide an efficient way to organize and manipulate data in a grid-like fashion, enabling a wide range of applications that involve multidimensional data representation and manipulation.