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};
  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. Intializing 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 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)

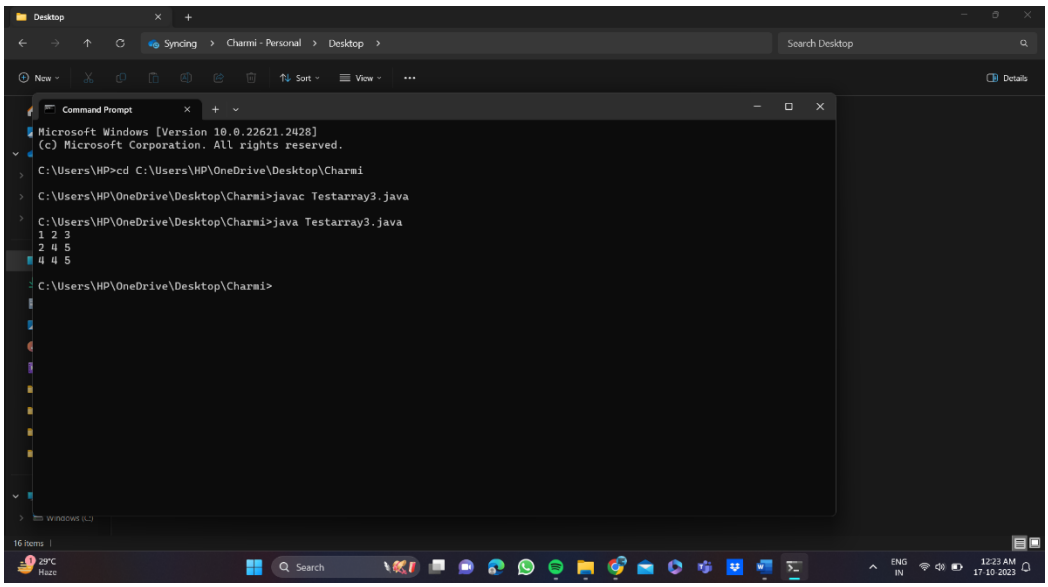
Example:

String demoString = new String ("GeeksforGeeks");

Code:

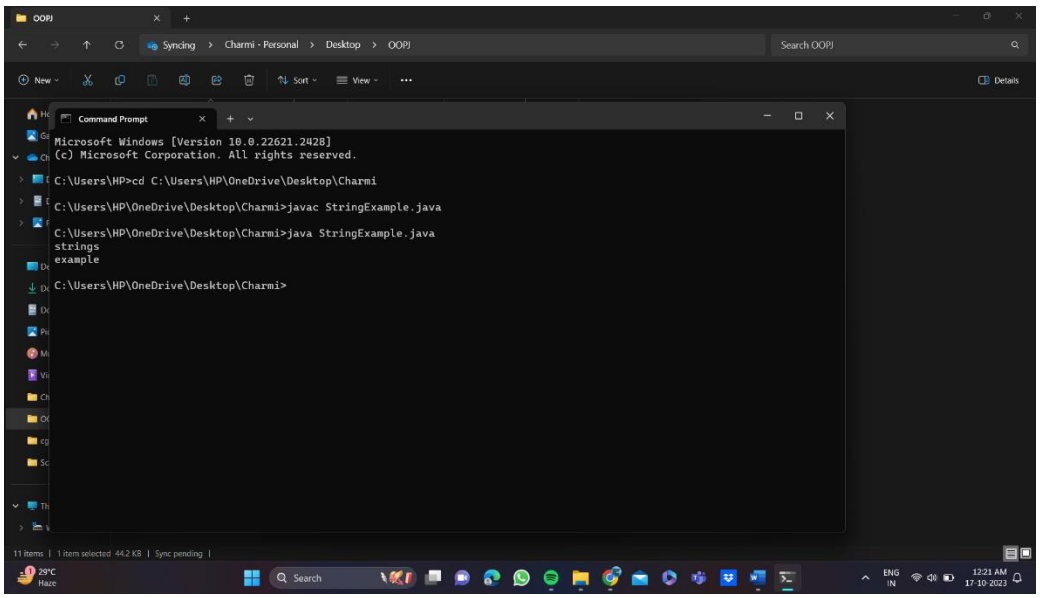
1}

```
class Testarray3{
public static void main(String args[]){
int arr[][]={{1,2,3},{2,4,5},{4,4,5}};
for(int i=0;i<3;i++){
 for(int j=0;j<3;j++){
   System.out.print(arr[i][j]+" ");
 }
 System.out.println();
}
}}
```



2}

```
class StringExample{
public static void main(String args[]){
String s1="java";
char ch[]={'s','t','r','i','n','g','s'};
String s2=new String(ch);
String s3=new String("example");
System.out.println(s2);
System.out.println(s3);
}}
```



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

String Usage:
1. `String s1 = "java";`: In this line of code, you have created a string variable `s1` and assigned it the value "java" using a string literal. String literals are enclosed in double quotes, and Java automatically creates a string object when a string literal is encountered. So, `s1` now holds the string "java."

2. `char ch[] = {'s','t','r','i','n','g','s'};`: Here, you've defined a character array `ch` with the values 's', 't', 'r', 'i', 'n', 'g', and 's'. Then, you create a string `s2` by passing the character array `ch` as an argument to the `String` constructor. This demonstrates creating a string from an array of characters.

3. `String s3 = new String("example");`: This line creates a string `s3` using the `new` keyword and the `String` constructor. You pass the string literal "example" as an argument to the constructor. This is another way to create a string object explicitly, as opposed to using string literals.

2D Array Usage:

1. `int arr[][] = {{1,2,3},{2,4,5},{4,4,5}};`: In this line, you have defined a 2D integer array `arr` with three rows and three columns. It represents a 3x3 grid of integer values. The double braces `{}` are used to initialize the array with specific values.

2. The nested loops in the `Testarray3` class: The nested `for` loops are used to iterate through the elements of the 2D array `arr`. The outer loop iterates over the rows, and the inner loop iterates over the columns. This allows you to access and print each element of the 2D array. The result is that the elements of the array are displayed, making it easier to visualize the 3x3 grid of integers. This code snippet demonstrates how to access and display elements from a 2D array by using nested loops.