



**Vidyavardhini's College of Engineering and Technology**

**Department of Artificial Intelligence & Data Science**

---

|  |
|--|
| 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)

**//Save by A.java**

```
package pack2;

public class A
{
    public static void msg()
    {
        System.out.println("Welcome to Package!!");
    }
}
```

**//Save by B.java**

```
package mypack;

import pack2.*;
```



```
class B
{
    public static void main(String args[])
    {
        A obj=new A();
        obj.msg();
    }
}
```

A screenshot of a Windows Command Prompt window. The title bar says 'Command Prompt'. The text inside shows the following commands and output:  
Microsoft Windows [Version 10.0.22000.1936]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users\parik>cd C:\Users\parik\OneDrive\Desktop\Priyanka Bhandari 02  
C:\Users\parik\OneDrive\Desktop\Priyanka Bhandari 02>javac ./pack2/A.java  
C:\Users\parik\OneDrive\Desktop\Priyanka Bhandari 02>javac B.java  
C:\Users\parik\OneDrive\Desktop\Priyanka Bhandari 02>java B.java  
Welcome to Package!!  
C:\Users\parik\OneDrive\Desktop\Priyanka Bhandari 02>

### Conclusion:

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

### String Usage:

String s1 = "java"; Here, we've created a string s1 using a string literal.



`char ch[] = {'s','t','r','i','n','g','s'};` We've defined a character array `ch`, and then we've created a string `s2` using this character array. This demonstrates the creation of a string from an array of characters.

`String s3 = new String("example");` This is another way to create a string, using the new keyword and a constructor. We have created `s3` from the string literal "example".

**2D Array Usage:**

`int arr[][] = {{1,2,3},{2,4,5},{4,4,5}};` We defined a 2D integer array `arr` with three rows and three columns. This represents a 3x3 grid of integer values.

The nested loops (for loops) in the `Testarray3` class are used to iterate through the elements of the 2D array and print them out. This demonstrates how to access and display elements from a 2D array.