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)

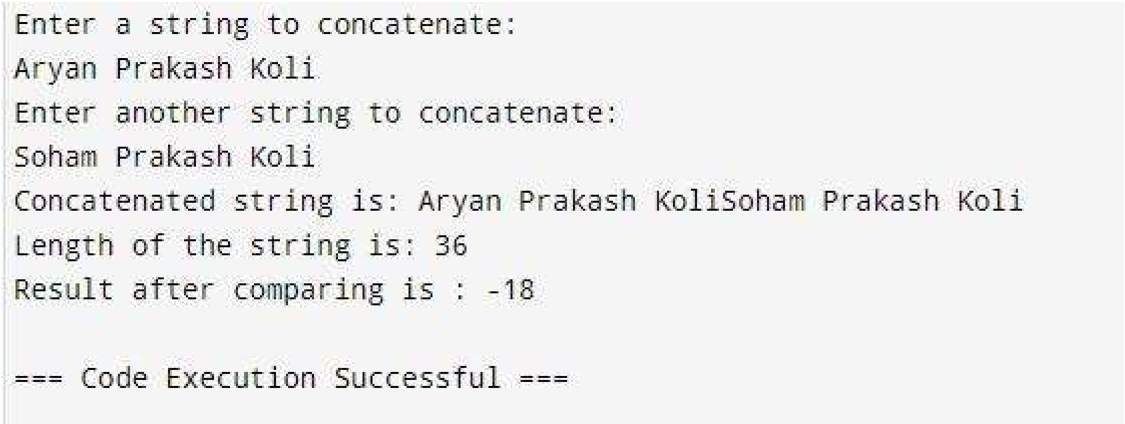
Example:

String demoString = new String (“GeeksforGeeks”);

Code for string functions:

```
import java.util.Scanner;
import java.io.*;
public class StringConcatenationExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);



        String predefinedString = " World";


        System.out.println("Enter a string to concatenate:");
        String userInput = scanner.nextLine();
         System.out.println("Enter another string to concatenate:");
        String userInput1 = scanner.nextLine();



        String result = userInput.concat(userInput1);
        System.out.println("Concatenated string is: " + result);
           int output=result.length();

        System.out.println("Length of the string is: " + output);
           int show=userInput.compareTo(userInput1);
            System.out.println("Result after comparing is : " + show);

        scanner.close();
    }
}
```

Output:

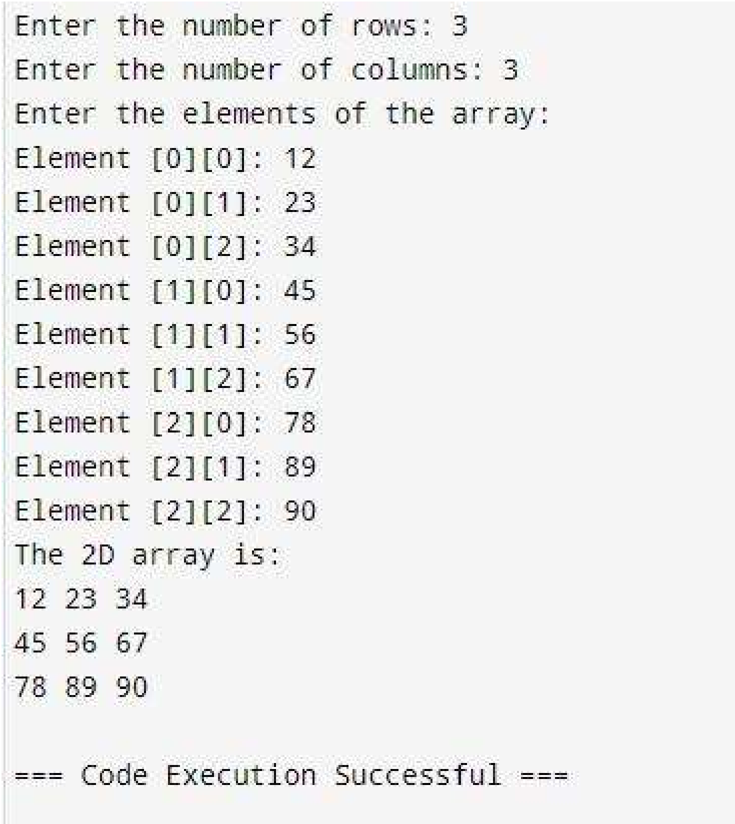


```
Enter a string to concatenate:
Aryan Prakash Koli
Enter another string to concatenate:
Soham Prakash Koli
Concatenated string is: Aryan Prakash KoliSoham Prakash Koli
Length of the string is: 36
Result after comparing is : -18

=== Code Execution Successful ===
```

Code for 2D arrays:

```
import java.io.*;
import java.util.Scanner;

public class TwoDArrayInputExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of rows: ");
        int rows = scanner.nextInt();
        System.out.print("Enter the number of columns: ");
        int columns = scanner.nextInt();

        int[][] array = new int[rows][columns];

        System.out.println("Enter the elements of the array:");

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                System.out.print("Element [" + i + "][" + j + "]: ");
                array[i][j] = scanner.nextInt();
            }
        }

        System.out.println("The 2D array is:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                System.out.print(array[i][j] + " ");
            }
            System.out.println();
        }

        scanner.close();
    }
}
```

Output:



```
Enter the number of rows: 3
Enter the number of columns: 3
Enter the elements of the array:
Element [0][0]: 12
Element [0][1]: 23
Element [0][2]: 34
Element [1][0]: 45
Element [1][1]: 56
Element [1][2]: 67
Element [2][0]: 78
Element [2][1]: 89
Element [2][2]: 90
The 2D array is:
12 23 34
45 56 67
78 89 90

=== Code Execution Successful ===
```

Conclusion:

In summary, the concepts of strings and 2D arrays in Java are integral for handling textual data and structured information. Strings provide powerful tools for manipulating and analyzing text, while 2D arrays enable the organization of data in a tabular format. Their combined usage allows for effective data processing and representation, making them essential in various programming tasks, including data analysis, user input handling, and grid-based applications. This synergy enhances the ability to build dynamic and responsive applications.