

# Solution to Practice Problems: Multi-dimensional Arrays

---

## 1. Tracing Programs

For each program below, show what is displayed on the screen when the code executes.

```
import java.util.Arrays;
public class Array2D
{
    public static void main(String [] args)
    {
        String [][] table = new String[2][3];
        table[0][0] = "team1";
        table[0][1] = "team2";
        table[0][2] = "result";

        table[1][0] = "Temple";
        table[1][1] = "USF";
        table[1][2] = "ugh";

        for(int i=0; i<table.length; i++)
        {
            System.out.println(Arrays.toString(table[i]));
        }

        System.out.println();
        for(int i=0; i<table.length; i++) {
            for(int j=0; j<table[i].length; j++) {
                System.out.print(table[i][j]);
                System.out.print(" ");
            }
            System.out.println();
        }
    }
}
```

### Screen

```
[team1, team2, result]
[Temple, USF, ugh]
```

```
team1 team2 result
Temple USF ugh
```

```

import java.util.Arrays;
public class Array2D2
{
    public static void main(String [] args)
    {
        double [][] values = { {3, 5.5, -7.2}, // row 1
                                {2, -2.5},      // row 2 (only 2 elements)
                                {1.5} };        // row 3 (only 1 element)

        for(int j = 1; j < values.length; j++) {
            for(int k = 0; k < values[j].length; k++) {
                values[j][k] = values[j-1][k];
            }
        }

        for(int j = 0; j < values.length; j++) {
            for(int k = 0; k < values[j].length; k++)
            {
                System.out.print(values[j][k] + " ");
            }
            System.out.println();
        }
    }
}

```

#### **screen**

```

3.0 5.5 -7.2
3.0 5.5
3.0

```

## **2. Writing Short Methods**

- a. Write a method that computes the sum of the numbers in an array and returns the sum.

```

public static int sum(int [] arr)
{
    int sum = 0;
    for(int i=0; i<arr.length; i++)
    {
        sum = sum + arr[i];
    }
    return sum;
}

```

- b. Write a method that computes the sum of the numbers in a 2D array and returns the sum.

```

public static int sum(int [][] arr) {
    int sum = 0;
    for(int i=0; i<arr.length; i++) {
        for(int j=0; j<arr[i].length; j++) {
            sum = sum + arr[i][j];
        }
    }
    return sum;
}

```

- c. \*Write a method that takes three 2D double arrays as arguments. The method should compute the matrix product of the first two arguments, and store it in the 3rd.

```
// this makes lots of assumptions about m1, m2, and result
// having the appropriate numbers of rows and columns
public static void matrixMultiply(
                                double [][] m1,
                                double [][] m2,
                                double [][] result)
{
    for(int row1=0; row1<m1.length; row1++) {
        for(int col2=0; col2<m2[0].length; col2++) {
            //initialize
            result[row1][col2] = 0.0;
            for(int k=0; k<m2.length && k<m1[row1].length; k++) {
                result[row1][col2] += m1[row1][k] * m2[k][col2];
            }
        }
    }
    // no return. why?
}
```

- d. Write a method that takes an array of Strings as an argument. It should create a new array of the same length, and copy the elements from the first array to the new one in reverse order. It should return the new array.

```
public static String [] getReverseArray(String [] arr)
{
    if(arr==null) { return null; } // to avoid NullPointerExceptions

    // first, create a variable to store the thing we want to return
    String [] ret = new String[arr.length];

    for(int i=0; i<ret.length; i++) {
        ret[i] = arr[arr.length-1-i];
    }
    return ret;
}
```