CSC 1051 – Exercises on Arrays of Objects and 2D Arrays

A) Arrays of objects
1. Review the textbook example of a DVD database: DVD.java, DVDCollection.java and Movies.java.

2. Aren’t shoes better than movies? Maybe you disagree, but, in any event, we will create a program similar to the one above, using shoes instead of DVDs. Using the Shoe class you designed for Project 5, create a class called ShoeCollection, similar to the DVDCollection – it should maintain a database of Shoe objects, using an array of Shoes. Create a program WalkInMyShoes.java to test ShoeCollection.java (WalkInMyShoes.java should be similar to Movies.java).

B) 2D Arrays
1. Draw a picture showing the array contents after execution of the following code fragments.

```java
int count = 1;
int[][] table = new int[2][3];
for (int i=0; i < 2; i++)
    for (int j=0; j < 3; j++)
    {
        table[i][j] = count;
        count++;
    }
```

**ANSWER:**

```
Table
```

Example

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

1 If you prefer, you may do this exercise using a different class, such as the Account class or the Person class – any kind of object could be used in place of DVD objects. The point is to use an array to store a collection of objects.
int[][] tableA = new int[3][3];
for (int i=0; i < 3; i++)
    tableA[i][i] = i;

char[][] tableB = new char[4][4];
String sample = "OPEN THIS FIRST";
for (int i=0; i < 4; i++)
    for (int j=0; j < 4; j++)
        tableB[i][j] = sample.charAt(i+j);

int[][] tableC = new int[3][4];
for (int i=0; i < tableC.length; i++)
    for (int j=0; j < tableC[i].length; j++)
        tableC[i][j] = i * 2 + j;

2. Create a test program to test each of the above code fragments and to show the array contents after execution. Use these programs to verify your answers to the previous question. Note that you will need to write some extra code to display the array contents (see, for example the boldfaced code below).

**EXAMPLE:**

```java
int count = 1;
int[][] table = new int[2][3];
for (int i=0; i < 2; i++)
    for (int j=0; j < 3; j++)
        {
            table[i][j] = count;
            count++;
        }

//print out the array contents
for (int i=0; i < 2; i++) /// processing rows
    {
        for (int j=0; j < 3; j++) //processing columns
            System.out.print( table[i][j] + " ");
        System.out.println(); // done with row, go to new line
    }
```
Additional Practice with 2D Arrays

1. Modify TwoDArray.java (Chapter 8) as follows:
   a) make it print the indices along the top (col) and left (row).
   b) ... and a box around the whole thing (use | and _ to create the box).

2. Write a method fill2D() that creates and returns a 2D array of dimensions n x m, where n, m are int parameters, and the array is filled with consecutive numbers starting at 1. For example, if invoked as fill2D(2,3) it should return the array:

<table>
<thead>
<tr>
<th>0 1 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3</td>
</tr>
<tr>
<td>4 5 6</td>
</tr>
</tbody>
</table>

3. Write a method with one int parameter n that creates and returns a two dimensional array of size n x n that consists of values of type double that are all set to zero, except the ones along the diagonal, which are all set to 1. So, for example, if the parameter is 3, the method should return an array with the following contents:

<table>
<thead>
<tr>
<th>0 1 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 0</td>
</tr>
<tr>
<td>0 1 0</td>
</tr>
<tr>
<td>0 0 1</td>
</tr>
</tbody>
</table>

4. Download, compile, and run SodaSurvey.java (Chapter 8). Modify the code to add static method computeAVG() that computes and returns the average of all the elements in a 2D array; the method should accept a single parameter, a two dimensional array of int, and it should return the average as a double. Use computeAVG() in the main() method of SodaSurvey.java to compute and print the average of all the soda scores (this will be the overall average, not by soda or by person, as computed in the program).

   http://www.csc.villanova.edu/~map/1051/Chap08/SodaSurvey.java