CSC 1051 - Lab 14

Objectives:
Practice using arrays of objects and two-dimensional arrays

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

2. Aren’t books better than movies? Maybe you disagree, but, in any event, we will create a program similar to the one above, using books instead of DVDs\(^1\). Using the Book class you designed for Project 6, create a class called BookCollection, similar to the DVDCollection – it should maintain a database of Book objects, using an array of Books. Create a program Lab14a.java to test BookCollection.java (Lab14a.java should be similar to Movies.java).

Have your work checked by the instructor or TA: _________________________________

(Hand in this worksheet after you complete the remaining questions.)

B) 2D Arrays
1. Show 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`

```
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</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[][] table = new int[4][4];
for (int i=0; i < 4; i++)
    table[i][i] = i;

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

int[][] table = new int[3][4];
for (int i=0; i < table.length; i++)
    for (int j=0; j < table[i].length; j++)
        table[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 - you can see how this is done in the example. You will need to adapt this code to print out the array contents of arrays for different sizes.

**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
}
```

Have your work checked by the instructor or TA: __________________________
**Additional Practice with Arrays**

We don’t have a project due this week, but these optional exercises will help you gain a stronger understanding of 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 \(n \times m\), 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:

   \[
   \begin{array}{ccc}
   0 & 1 & 2 \\
   1 & 2 & 3 \\
   4 & 5 & 6 \\
   \end{array}
   \]

3. Write a method with one int parameter \(n\) that creates and returns a two dimensional array of size \(n \times n\) 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:

   \[
   \begin{array}{ccc}
   0 & 1 & 2 \\
   1 & 0 & 0 \\
   0 & 1 & 0 \\
   0 & 0 & 1 \\
   \end{array}
   \]

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).