TCSS143
Fundamentals of Object-Oriented Programming-Theory and Application
Programming Assignment 2

DUE: Tuesday, October 7, 2014 by 11:59 p.m.

The purpose of this programming project is to demonstrate understanding and use of 2D Arrays, File Input/Output (I/O), try/catch blocks, and the basic concept of exceptions.

This program will perform a basic multiplication of two matrices. The rules for matrix addition, subtraction, and mulitplication are clearly described at:

http://www.algebralab.org/lessons/lesson.aspx?file=Algebra_matrix_operations.xml
(Algebra Lab website of Mainland High school. Mulitiplication is found about 2/3 down the page below the topic of scalar multiplication. Also, there is a plethora of websites that describe the process)

Your program will read data from an input file (in2.txt) into the 2 arrays to be multiplied, multiply the 2 arrays and assign the results to a 3rd array, and then display the contents of all 3 arrays in an output file (out2.txt). Each input array set of numbers will be preceded by 2 integers; the first for the row dimension and the second for the column dimension as illustrated below (all data will be integers):

```
3  4
 2  4  1  7
-1  0 -2  2
 3 -4  5  3

4  5
 3   5  -9  12 -10
-3   2 15  -4   0
 1   0  -7   1   2
 5   3   7  -2  15
```

The first array is a 3x4 and the second is a 4x5. Based on the rules of matrix multiplication, the resulting array will be a 3x5 array:

```
  30   39    84   -5   87
   5    1    37  -18   36
  41   16  -101   51   25
```

You may assume the input data is valid and the 2 arrays meet the requirements of matrix mulitplication.

Your main method will open the files for I/O, create and populate the 2 arrays with the data from the input file (in2.txt), generate a multiplication result array and produce clear, meaningful output (out2.txt) to display:

- ✓ **the 2 arrays on which the mulitplication is being performed**
- ✓ **the multiplication result array**

You should name your class Assign_2 (file name will be Assign_2.java). You should also create at least 3 methods:

- ✓ **one that is passed the input file Scanner and returns a 2D array.**
- ✓ **one that multiplies the 2 input arrays and returns a 2D array result.**
- ✓ **one that displays the arrays to an output file.**

You should **NOT** use throws FileNotFoundException but instead open your files with try/catch blocks as follows:

```
// Documentation, imports, class header, etc. goes here.
public static void main(String[] theArgs) {
    Scanner input = null;
    PrintStream output = null;
    String inFileName = "in2.txt";
    String outFileName = "out.txt";
    boolean filesOk = false; // Indicates if the files are accessable.

    try {
      input = new Scanner(new File(inFileName));
      output = new PrintStream(new File(outFileName));
      filesOk = true;
      }
    catch (FileNotFoundException e) {
      System.out.println("Can't open file - " + e);
    }
    if (filesOk) {
     ...  // in this block is where the rest of the processing should
          // occur.
     }
   } // End of main.
```

Upload Assign_2.java on Canvas by the due date/time.

Several sample output file results based on various input files are listed:

```
Array A:
  2  4  1  7
 -1  0 -2  2
  3 -4  5  3

        X

Array B:
   3   5  -9  12 -10
  -3   2  15  -4   0
   1   0  -7   1   2
   5   3   7  -2  15

        =

Array C:
   30   39   84   -5   87
    5    1   37  -18   36
   41   16 -101   51   25
```

```
Array A:
  3  4 -2  7
 -1  0 -2  2
  3 -4  5  3

        X

Array B:
  3461     5    -9    12   -10
    -3     2    15    -4     0
     1     0  -781     1     2
     5     3     7    -2    15

        =

Array C:
  10404     44   1644      4     71
  -3453      1   1585    -18     36
  10415     16  -3971     51     25
```

One More Exampe Next Page ---->

```
Array A:
  3  4 -2  7  2
 -1  0 -2  2  1
  3 -4  5  3  3
 -1  0  1 -2 -1

        X

Array B:
   3   5  -9  12 -10   5
  -3   2  15  -4   0   1
   1   0  -7   1   2   4
   5   3   7  -2  15   0
   2   0   1   5   2  10

        =

Array C:
  34  44  98  14  75  31
   7   1  38 -13  38  -3
  47  16 -98  66  31  61
 -14 -11 -13 -12 -20 -11
```