CSCI 3302 Programming Assignment 01 (100 Points)

Due: Sept 5 12:00 AM

OBJECTIVES:

- Demonstrate a basic understanding of Java programming.
- Demonstrate basic competency in Java text file IO.
- Demonstrate understanding of the double arrays
- Demonstrate understanding of using Git/GitHub for assignments.

ASSIGNMENT ASSISTANCE:

- This homework assignment is due before the date and time specified above.
- This assignment is restricted to individual effort. As per our syllabus, the use of AI is prohibited. You may not receive help from any other person except the instructor, the TA, or the AARC (help from the TA and AARC must be well documented!).
- Any resource used (other than Dr. Becnel or the course text) must be documented in the code (as comments) detailing the source and describing exactly what was learned and how that information was used. Submissions will be severely penalized if copied in part or whole from any source.
- If you need help, visit your instructor during his posted office hours. If your schedule cannot accommodate any of these times, then email your instructor to schedule a different time.

PROBLEM DESCRIPTION:

1. Create a Java class called `Table` in a class file called `Table.java`.

2. The `Table` class contains a single attribute; a double array named `values`.

3. The `Table` class contains a constructor and five methods as members of the `Table` class:

    a. The lone `public` constructor takes two arguments, the number of rows and the number of columns in the table. The constructor should instantiate the `values` attribute.
        i. An `IllegalArgumentException` with an appropriate message should be thrown if either argument is 0 or negation.

    b. Two `public` methods called `getNumberOfRows, getNumberOfColumns` return an integer representing the number of rows and columns respectively.

    c. A `public` method called `getValue` that takes two arguments of type `int` called `row, column` and returns a value from the `values` attribute at the given row and column value. If the index is out of the allowable range, then the method throws the `IllegalArgumentException`.

        i. Note: arrays are indexed at 0; however individuals who use Tables start counting at 1, so you need to make the necessary adjustment to the given values

d. A `public` method called `setValue` that takes three arguments: two of type `int` called `row, column` and one of type double call `value`. The method sets a value from the `values` attribute at the given row and column value. If the index is out of the allowable range, then the method throws the `IllegalArgumentException`.

e. Two `public` methods called `getRowMax, getRowMin` that takes one argument of type `int` called `row` and return the largest and smallest values on the given row, respectively. If the index is out of the allowable range, then the method throws the `IllegalArgumentException`.

f. Two `public` methods called `getColumnMax, getColumnMin` that takes one argument of type `int` called `column` and return the largest and smallest values on the given column, respectively. If the index is out of the allowable range, then the method throws the `IllegalArgumentException`.

g. A `public` method called `getRowAverage`, that takes one argument of type `int` called `row` and returns the average of the values on the given row. If the index is out of the allowable range, then the method throws the `IllegalArgumentException`.

h. A `public` method called `getColumnAverage,` that takes one argument of type `int` called `column` and returns the average of the values on the given column. If the index is out of the allowable range, then the method throws the `IllegalArgumentException`.

i. Two `public` methods called `getMax, getMin` that return the largest and smallest values in the table, respectively.

j. One `public` method called `getAverage` that returns the average of all values in the table, respectively.

k. One `public` method called `getNumberInRange` that takes two arguments of type `double` called `a` and `b`. The method returns an integer representing the number of values in the table that are in between a and b, inclusive. Note: you may NOT assume that $a \leq b$.

l. A `public` method called `loadValues`. This method takes a single attribute called `filename` of type `String`. The method will throw the `FileNotFoundException`. The method reads a series of double values from a text file and stores them in the `values` attribute. The number of values read corresponds to the number of rows/columns in the table. Here is an example of the file:

```
1.2 2.3 3.4 5.6
2.1 4.3 5.4 6.5
3.1 5.3 6.4 1.5
```

The code for this method is:

```java
public void loadValues(String filename) throws FileNotFoundException {
    File dataFile = new File(filename);
    Scanner scanner = new Scanner(dataFile);
    for (int r = 0; r < this.getNumberOfRows(); r++)
        for (int c = 0; c < this.getNumberOfColumns(); c++)
            this.values[r][c] = scanner.nextDouble();
    scanner.close();
}
```

m. A `public` method called `toString`. This method returns the string representation of the Table. For example, the string representation of the above table is simply

```
1.2 2.3 3.4 5.6
2.1 4.3 5.4 6.5
3.1 5.3 6.4 1.5
```

4. The file `Table.java` should NOT contain a main method. You can include files with testing code; however, these will not be considered for grading. If you wish to include non-working code for insight into your thought process, make sure to contain it within comment blocks and ensure that the submission successfully compiles.

5. Your program should work in the GitHub codespace (Linux environment) and locally (Windows environment).

SUBMISSION:

- Review the Evaluation below to ensure you have met all the requirements.
- Commit `Table.java` to GitHub. Upload a backup copy to D2L.

EVALUATION

Remember to consult Program Requirements.docx

▪ **Automatic Deductions:**	
Late/Not Submitted	-100
Code not submitted to GitHub	-30
Code does not run/compile	-50
Earn Points for the following:	
Code has a comment header with name, section, date	5 pts
Code organization, structure, and indention is appropriate (SHFT + ALT + F in VS Code)	5 pts
Code is well and meaningfully commented.	5 pts
Appropriate variable and method names that follow Java conventions	5 pts
Instructions correctly followed for fields, class, methods	10 pts
Constructor	5 pts
getNumberOfRows/Columns get/setValue	5 pts
getRow/ColumnMax/Min	10 pts
getRow/ColumnAverage	10 pts
getMax/Min	10 pts
getAverage	5 pts
getNumberInRange	10 pts
loadDataFile	10 pts
toString	5 pts