

Montgomery College
CMSC 203
Assignment 5 Design

Write up included below

TwoDimRaggedArrayUtility
+getTotal() +getAverage() +getRowTotal() +getColumnTotal() +getHighestInRow() +getHighestInRowIndex() +getHighestInColumn() +getHighestInColumnIndex() + getLowestInColumn() + getLowestInColumnIndex() + getHighestInArray() + getLowestInArray()

HolidayBonus
+calculateHolidayBonus() +calculateTotalHolidayBonus()

1) Write the pseudo code for the methods of *TwoDimRaggedArrayUtility* and *HolidayBonus* class based on the Assignment 5 Description given to you. Refer to the [Pseudocode Guideline](#) on how to write Pseudocode.

TwoDimRaggedArrayUtility

readFile()

receives file

returns file contents in two dimensional arrays

`writeToFile()`

writes two-dimensional array to file

prints each row on a separate line

`getTotal()`

receives two-dimensional array of doubles

returns sum of elements in array

`getAverage()`

receives two-dimensional array of doubles

returns average of elements in array

`getRowTotal()`

receives two-dimensional array of doubles

returns sum of a row

`getColumnTotal()`

receives two-dimensional array of doubles

returns sum of elements in a column

`getHighestInRow()`

receives two-dimensional array of doubles

returns highest value in row

`getHighestInRowIndex()`

receives two-dimensional array of doubles

returns index of highest value in row

`getLowestInRow()`

receives two-dimensional array of doubles

returns lowest value in row

`getLowestInRowIndex()`

receives two-dimensional array of doubles

returns index of lowest value in row

`getHighestInColumnIndex()`

receives two-dimensional array of doubles

returns largest element in column

getHighestInColumn()

receives two-dimensional array of doubles

returns index of highest value in column

getLowestInColumn()

receives two-dimensional array of doubles

returns lowest value in column

getLowestInColumnIndex()

receives two-dimensional array of doubles

returns index of lowest value in column

getHighestInArray()

receives two-dimensional array of doubles

returns highest value in array

getLowestInArray()

receives two-dimensional array of doubles

returns lowest value in array

Holiday Bonus

calculateHolidayBonus()

receives two-dimensional array of doubles, bonus amount for highest selling/lowest selling stores

returns doubles for each store in the district

calculateTotalHolidayBonus()

receives two-dimensional array of doubles, bonus amount for highest sales category/lowest sales category, and bonus for all other stores.

returns doubles for all Holiday Bonuses for the district

2) Complete the following test table. At this point you only need to complete the **Input** and **Expected Output** columns. Later when the implementation is complete, you will complete the **Actual Input** and **Actual Output** columns and compare them to see if the tests passed or not.

Make sure your tests cover all the possible scenarios.

Data Set1

Test Case #	Input	Actual Input	Expected Output	Actual Output	Did the test pass?
1	1 2 3	1 2 3	\$1.00 \$2.00 \$3.00	\$1.00 \$2.00 \$3.00	yes
2	4 5	4 5	\$4.00 \$5.00	\$4.00 \$5.00	yes
3	6 7 8	6 7 8	\$6.00 \$7.00 \$8.00	\$6.00 \$7.00 \$8.00	yes

Pseudocode Guideline

Pseudocode is code written for human understanding not a compiler. You can think of pseudocode as “English code,” code that can be understood by anyone (not just a computer scientist). Pseudocode is not language specific, which means that given a block of pseudocode, you could convert it to Java, Python, C++, or whatever language you so desire.

Pseudocode will be important to your future in Computer Science. Typically pseudocode is used to write a high-level outline of an algorithm.

As you may already know, an algorithm is a series of steps that a program takes to complete a specific task. The algorithms can get very complicated without a detailed plan, so writing pseudocode before actually coding will be very beneficial.

How to Write Pseudocode

There are no concrete rules that dictate how to write pseudocode, however, there are commonly accepted standards. A reader should be able to follow the pseudocode and hand-simulate (run through the code using paper and pencil) what is going to happen at each step. After writing pseudocode, you should be able to easily convert your pseudocode into any programming language you like.

We use indentation to delineate blocks of code, so it is clear which lines are inside of which method (function), loop, etc. Indentation is crucial to writing pseudocode. Java may not care if you don't indent inside your **if** statements, but a human reader would be completely lost without indentation cues.

Remember: Human comprehension is the whole point of pseudocode. So, what does pseudocode look like?

Pseudocode	Real Code in Java
------------	-------------------

Declare an integer variable called n Declare an integer variable sum. Declare an integer variable f1 Declare an integer variable f2 If n is less than 2 sum = n else set sum to 0 set f1 and f2 to 1 repeat n times sum = f1 + f2 f2 = f1 f1 = sum end loop print sum	<pre> int n,k, f1, f2, sum; if (n < 2) sum =n; else { sum=0; f1 = f2 = 1; for(k=2; k<n; k++) { sum = f1 + f2; f2 = f1; f1 = sum; } } System.out.println("Fibonacci of number " + n + " is " + sum); </pre>
---	---

Finding the Fibonacci numbers till n:

Remember that pseudocode is not language specific so we are not looking for “almost Java” code, but instead, we are looking for a strong understanding of the algorithm at hand.

Learning Experience: highlight your lessons learned and learning experience from working on this project.

- What have you learned?
 - I learned that reading from files can be incredibly difficult. Storing data into ragged arrays was very complicated.
- What did you struggle with?
 - I struggled writing the Unit Tests and figuring out how to read the different files.
- What will you do differently on your next project?
 - I’ve learned the importance of working through certain procedures such as looping, reading files, and writing files repeatedly. Knowing how to implement something isn’t enough.
- Include what parts of the project you were successful at, and what parts (if any) you were not successful at.
 - I’ve learned to get things up and running fairly quickly. I was not successful and getting the highest sales to highlight.

Assignment 5 Check List (include Yes/No or N/A for each item)

#		Y/N or N/A	Comments
	Assignment files:		
	<ul style="list-style-type: none"> FirstInitialLastName_Assignment5_Moss.zip 	Yes	
	<ul style="list-style-type: none"> FirstInitialLastName_Assignment5_Complete.zip 	Yes	
	Program compiles	Yes	
	Program runs with desired outputs related to a Test Plan	Yes	
	Documentation file:		
	<ul style="list-style-type: none"> Comprehensive Test Plan 	Yes	
	<ul style="list-style-type: none"> Screenshots for each Junit Test 	Yes	
	<ul style="list-style-type: none"> Screenshots for each Test case listed in the Test Plan 	Yes	
	<ul style="list-style-type: none"> Screenshots of your GitHub account with submitted Assignment# (if required) 	Yes	
	<ul style="list-style-type: none"> UML Diagram 	Yes	
	<ul style="list-style-type: none"> Algorithms/Pseudocode 	Yes	
	<ul style="list-style-type: none"> Flowchart (if required) 	Yes	
	<ul style="list-style-type: none"> Lessons Learned 	Yes	
	<ul style="list-style-type: none"> Checklist is completed and included in the Documentation 	Yes	