WESTERN UNIVERSITY
DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

ES1036B
PROGRAMMING FUNDAMENTALS FOR ENGINEERS
WINTER 2022

Lab08 – Part 1: 2D Arrays & File I/O (ET2+KB4)

Instructor:
DR. QUAZI RAHMAN

Prepared by:
HIRA NADEEM

Lead TA: Hira Nadeem, hnadeem5@uwo.ca

1 Goal

In this lab, we will explore 2D arrays and File I/O in Java. Two dimensional arrays expand the capabilities of arrays in Java to incorporate more data. They are stored as arrays of arrays and have both rows and columns. Arrays can also be used with File I/O in Java. File I/O is used to perform read and write operations on files stored in local storage.

2 Resources

Lecture Notes:	Unit 6: Arrays and Files, Unit 5: Control Statements Refresh Unit 4: Class Fundamentals in Coding, Unit 3: Methods in Java
Pre-recorded lecture videos:	Session 3: 2D Array, Session 4: Files
IntelliJ Tutorials:	Arrays and Methods, Array of Class, 2D Arrays, Files

3 Deliverables

One zip file folder containing your project folder. Name it username_Lab8.zip where username is the beginning part of your email (e.g., Dr. Rahman's email address is qrahman3@uwo.ca, and his UWO username is: qrahman3).

Submission deadline: Submit your code by the end of your lab session during the Week of April 4 – 8, 2022. Prepare to demonstrate your understanding during the lab session.

4 Good Programming Practices

Below are several programming practices that should be followed to write and maintain quality codes. Please note that you will be marked on all these components:

- Include meaningful comments in your program. This will help you remember what each part of your code does, especially after long breaks from your work. Your TA will also appreciate understanding your code by going over your comments.
- Choose meaningful and descriptive names for your variables. There is a balance between descriptive names and code readability but always err on the side of descriptive.
- It is recommended that you follow the naming strategy for variables, methods and class names, as outlined in your course handout (Unit 2). Since the identifiers cannot contain white-space characters (spaces or tabs), words in an identifier should be separated by uppercase letters (myNewFunction(), myNewVariable). For class names, capitalize the first letter of each word in the name (e.g. MyClass, MatrixCalculator). For any constant name, use uppercase letters, and if needed, concatenate two or more words with underscore (e.g., MINIMUM_DRIVING_AGE)
- Initialize variables when declaring them. This means giving them initial values which are easier to track in your program if logical errors are present with your output.
- Indent and properly format your code. You should write your codes so that your teaching assistant can read and understand your code easily.
- Include a header in each of your java class files. The header should include your full name, UWO ID number, date the code was written and a brief description of the program in that file. Use any print statement to print the Header information on the screen based on the format below:

```
/*********************************************
* Add your full name here*
* Add your student number*
* Add Date*
* Give a brief description of the task *
*********************************************/

public class MyClass
{
    public static void main(String args[])
    {
      // Your code here
    }
}
```

5 Lab Assignment Questions

You will be using methods from your **MyMethod** class.

1. Under the current project's source (src) file, copy-paste your MyMethod Class along with its package.
2. Now, you can import the above package that contains MyMethod Class to the destination package.
3. Import the MyMethod class using the import statement at the top. This statement will look like this: **import thePackageName.MyMethod**; or **import thePackageName.***;
   Here the package name should be the name of the package where you have created **MyMethod** class.

Use IntelliJ IDE to create a project named ***username_Lab8***. This is Part 1 of 2, you will include Part 2 in this same project.

5.1 Question 1 (10 marks)

In this exercise, we will write the code to multiply two matrices. Application based scripting-language such as MATLAB provide the option to simply multiply the matrices with the multiplication operator and return the result. Internally, this code is written in C and added to MATLAB. In this exercise, we will write this code in Java. Here you will create a driver class called **TwoDArrayMultiplication_YourFirstName**, and define the following public static methods inside. You will also need to use methods from your **MyMethod** class.

1. Create a package named **Q1.**
2. Import your **MyMethod** class. (Instructions at the top of page 3 of this handout)
3. Create a public class called **TwoDArrayMultiplication_YourFirstName** with the following members:
   a. Method with header:
      `public static void populate2DArrays(int[][] ma)`
      The above method will populate a 2D integer array referenced by ***ma***. The array element values will be generated randomly between 2 and 5.
   b. Method with header:
      `public static void display2DArrays(int[][] ma)`
      The above method will display the 2D array elements referred to by ***ma***, as shown in the sample output. For proper formatting you can use **printf()** method with "%3d" format specifier for each data item.
   c. Method with header:
      `public static int[][] multiplyArrays(int[][] a, int[][] b)`
      The above method will multiply two 2D arrays referred to by a and b, and return a 2D array reference of $a \times b$. Multiplying two 2D array work on the following algebraic expression to find out the resultant array:

$$result[i][j] = \sum_{k=0}^{N} (a[i][k] * b[k][j])$$

      Where $N = b.length - 1$.

Hint: In this case, inside the method, declare a 2D **int[][]** array reference variable and instantiate it with a size of $a.length \times b[0].length$. They complete the rest using the series multiplication. This is an application of nested loop. The outermost loop will run from 0 to N. The middle loop will run for index ***i*** and the innermost one will run for index ***j***. Please see your instructor ASAP, if it does not make any sense.

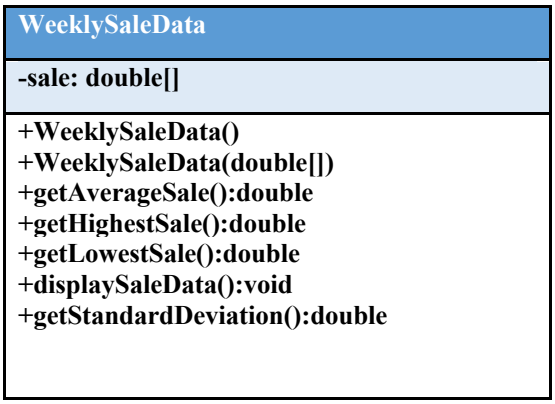
4. Driver method Specifications:
    a. Call the **MyHeader** method from the **MyMethod** class.
    b. With the help of an instantiated **Scanner** type reference variable ask the user to enter the row size, followed by the column size of the first array (see the sample output).
    c. Now ask the user to enter the row size of the second array. Validate that the entered value is equal to the column size of the first array. You are required to use a loop for validation.
    d. Now ask the user to enter the column size of the second array.
    e. Now declare these two arrays with the above entered size values.
    f. With the help of the **populate2DArrays()** method populate both arrays (see Unit 6 (extended version) slide 138).
    g. With the help of the **display2DArrays()** method write the appropriate statements to display both arrays on the screen (see the sample output) (see Unit 6 (extended version) slide 139).
    h. Now call the **multiplyArrays()** with the first-array and the second-array as arguments in the same order and assign the returned value to an 2D **int[][]** type array reference variable.
    i. At this point, with the help of **display2DArrays()** method, write the appropriate statements to display the resultant array.
    j. Call your **myFooter()** method from your **MyMethod** class.

Sample Output (on next page)

```
Note: Sample output from one of the runs. Since the numbers are generated randomly,
the result will be different in each run

*****************************************************
   Quazi Rahman
   Lab 8, Question 1
*****************************************************
Multiplying two 2D arrays:
Enter array1 info:
   Enter row-size: 2
   Enter column-size: 3
Enter array2 info:
   Enter row-size: 2
   Row size of the 2nd matrix has to be equal to the column size of first matrix!
   Enter again: 4
   Row size of the 2nd matrix has to be equal to the column size of first matrix!
   Enter again: 3
   Enter column-size: 5

Elements of Array 1, Size[2 x 3]:
2  5  5
4  4  3

Elements of Array 2, Size[3 x 5]:
3  3  2  3  3
3  5  5  2  3
4  5  5  5  5

Multiplying two arrays [Array 1 x Array 2]..
The resultant size is: [2 x 5].
The elements are...
41 56 54 41 46
36 47 43 35 39

*** Signing off - Quazi Rahman! ***
```

5.2 Question 2 (10 marks)

In this exercise, we will demonstrate our understanding by reading data from a given text file (Datafile_Lab8.txt) into an array, and after processing those data items in an array, we will update the Datafile_Lab8.txt. Please download the text file and save it to folder in your drive. Here you need to define two classes: **WeeklySaleData_YourFirstName**, and the driver class called **WorkingWithFilesAndArrays**. You will also need to use methods from your **MyMethod** class.



5. Create a package named **Q2.**
6. Import your **MyMethod** class. (Instructions at the top of page 3 of this handout)
7. Create a public class called **WeeklySaleData_YourFirstName** with the following members:
    a. A private field of double[].
    b. A constructor method without any argument that assigns null to the private field.
    c. A constructor with a double[] type array parameter.
    d. A public method **getAverageSale()** will return the average weekly sale in dollars.
    e. A public method **getHighestSale()** will return the highest weekly sale in dollars.
    f. A public method **getLowestSale()** will return the lowest weekly sale in dollars.
    g. A public method **displaySaleData()** will display the weekly sale values in dollars as shown in the sample output.
    h. A public method **getStandardDeviation()** will return the standard deviation of the weekly sale.
    As you already know, standard deviation (a frequently used term in Engineering and Science) is a number used to tell how measurements for a group of data items are spread out from the average (mean or expected value).
    A low standard deviation means that most of the numbers are close to the average, while a high standard deviation means that the numbers are more spread out. The standard deviation (σ) is defined as $\sigma = \sqrt{\frac{\sum_0^{N-1}(x_i-\mu)^2}{N}}$ where μ= mean and N= length of the array.
8. Create the class **WorkingWithFilesAndArrays** and add the driver method with the following specifications:
    a. Call the **MyHeader** method from the **MyMethod** class.
    b. Open the text file for reading, using the **File** type reference variable, referencing the given text file. You need to use the path information to access the file as we discussed and

demonstrated it in the class on the 18th of March (See Unit 6 slides 169-170). The text file contains the sale values of a week for a small business.

c. Create a **Scanner** type reference variable to read data from the file (See Unit 6 slides 171-173)
d. Read each data item into an array of size 7 (see Unit 6 slide 179)
e. Close the file (see Unit 6 slides 171, 180).
f. Create a **WeeklySaleData_YourFirstName** type reference variable and instantiate with the reference array variable you created above in d).
g. Now with the help of the reference variable created in f), display the array content using the **displaySaleData()** method (see the sample output). Also, print the highest, lowest, average, and standard deviation using the appropriate methods of the **WeeklySaleData_YourFirstName** class (see the sample output).
h. Open the text file again to update it with the help of **PrintWriter** and **FileWriter** classes (see slide 160-167).
i. Update the file with all the information you displayed in g) (see the content of the file below the sample output).
j. Close the file (see Unit 6 slides171, 180).
k. Call your **myFooter()** method from your **MyMethod** class.

Note: Make sure to include the descriptive print statements that show the order of the operations for each File I/O task like in the sample output.

Sample Output (on next page)

```
**************************************************
   Quazi Rahman
   Lab 8, Question 2
**************************************************
Reading the file...
Here is this week's sale-data...
[850.64, 274.64, 179.86, 910.13, 334.22, 943.66, 869.33]
Highest weekly Sale: 943.66
Lowest weekly Sale: 179.86
Average weekly Sale: 623.21
Standard Deviation of Weekly Sale: 315.98
.. Updating the file with the above info...
Closed the file.....

*** Signing off - Quazi Rahman! ***

*/
/*

//Note: The below content is not displayed, this is just how the file
//should look after it has been updated.
Content of the file Datafile_Lab8.txt
850.64
274.64
179.86
910.13
334.22
943.66
869.33

Highest weekly Sale: 943.66
Lowest weekly Sale: 179.86
Average weekly Sale: 623.21
Standard Deviation of Weekly Sale: 315.98
```

6 Lab Assignment Questions

Zip the project file, name it ***username_Lab8.zip*** and submit in OWL by the end of your lab session during the week of April 4-8, 2022. You will submit Part 2 in this same project folder.

If you have any questions about the Lab Handout, please contact the Lead TA, Hira Nadeem at hnadeem5@uwo.ca