# Lab Assignment 7\_2 - Arrays

## Part 1:

1. For this program you will be reading in a file of the daily high temperature for each day in a specified month and determining the lowest, highest and average temp for that month.
2. Create a new Java project called **Lab7\_2A** and a class named **Lab7\_2A**
3. Create a second class called **MonthTemperatures**
4. The **MonthTemperatures** class should have the following instance variables & methods
   1. An ArrayList of doubles named **tempArray** Only put the first half of the ArrayList declaration here 🡪 **ArrayList <Double> tempArray;**   
      Put the second half in the constructor 🡪 **tempArray = new ArrayList<Double>();**
   2. A string variable named **textFileName**
   3. Two integer variables named **month** & **year**
   4. Three double variables named **highest**, **lowest**, & **average**
   5. A constructor that receives a string parameter and assigns it to **textFileName**.  
      It should also set **highest** and **average** to 0 and **lowest** to 999
   6. A void method named **fillArray** that will read data from the text file (textFileName)
      1. It should read the month and year from the file
      2. Next it should use a loop to read double values until it hits the end of the file, and it should add each value to the **tempArray** ArrayList.
      3. After the loop print **tempArray** using the ArrayList.toString method (not the one we create here.)
      4. Call **findHighest**
      5. Call **findLowest**
      6. Call **computeAverage**
   7. A void method named **findHighest** that sets the instance variable **highest** equal to the biggest value in **tempArray**.
   8. A void method named **findLowest** that sets the instance variable **lowest** equal to the smallest value in **tempArray**.
   9. A void method named **computeAverage** that sets instance variable **average** equal to the average of the values in the array. (The array does not have a specified number of values in it, so you will need determine how many items are in **tempArray**.)
   10. A String **toString** method that returns a String of all the instance variables (except for tempArray & textFileName with labels. (As we have done in previous lab assignments.)
   11. Make all the instance variables private and the methods public. The methods will not have parameters unless the instructions list them.
5. Back in the main method
   1. Declare and instantiate a **MonthTemperatures** object named **month1** and send “Lab7\_2A1.txt”) as the parameter to its constructor.
   2. Call **fillArray** for **month1**
   3. Print **month1** using the **toString** shortcut
   4. Declare and instantiate a **MonthTemperatures** object named **month2** and send “Lab7\_2A2.txt”) as the parameter to its constructor.
   5. Call **fillArray** for **month2**
   6. Print **month2** using the **toString** shortcut

## Part 2:

1. For this program you will be creating grocery item objects, putting them into an array of objects, and working with them.
2. Create a new Java project called **Lab7\_2B** and a class named **Lab7\_2B** with your main method in it.
3. Create a second class in the project named **Item**
4. The **Item** class should have the following
   1. 4 instance variables
      1. itemName (String)
      2. itemPrice (double)
      3. itemQuantity (double)
      4. itemTotalCost (double)
   2. A constructor that receives inName (String), inPrice (double) and inQty (double) as parameters and fills in the instance variables with the parameter values  
      It should also call **computeCost**.
   3. A void method named **computeCost** that calculates itemTotalCost by multiplying itemQuantity by itemPrice
   4. A double method named **getTotalCost** that returns itemTotalCost
   5. A String method named **getItemName** that returns itemName.
   6. A String **toString** method that returns a String of all the instance variables with labels. (As we have done in previous lab assignments.)
   7. Make all the instance variables private and the methods public. The methods will not have parameters unless the instructions list them.
5. Back in the main method
   1. Create an array named cartArray of 4 **Item** objects.
   2. For each element in cartArray
      1. Read the item name, price, and quantity from the text file (“Lab7\_2B.txt”). (You will want to have local variables to hold these.)
      2. Instantiate the object sending those 3 pieces of data as parameters.
      3. Print the object using the **toString** shortcut.
      4. Get the total cost for that item (using the **getTotalCost** method) and add it to a local variable named subtotal
   3. Calculate the tax for the whole order by multiplying subtotal by 0.0825
   4. Calculate the total for the order by adding subtotal to tax.
   5. Print a blank line and then print subtotal, tax, and total each on their own line with a label.
   6. Use a new loop to go through the array and find the item with the highest itemTotalCost.
   7. After the loop print a blank line and then the highest itemTotalCost and its item name (with labels as usual.)

\*Note – Print all monetary amounts (price, tax, subtotal, etc. with 2 decimal places)

Example: You can do this by replacing **itemPrice** with **String.format("$%.2f",itemPrice)** when you print it or add it to toString’s return value.

\*\*\* As you are experiencing, the programs get more involved as we go along, but remember that I am here to help if you get stuck.