

Assigned Date: 4/4

Due date: 4/12 or 4/19 due to the spring break (4/11 – 4/17)

DO NOT share your answers with anyone. DO NOT collaborate on completing work with anyone. DO NOT use the Internet to search for specific solutions to assignments. DO NOT pay anyone to complete your assignment. **Avoid web sites that offer solutions to assignments. If you copy work from such web sites, keep in mind that other students are also looking at the same information and will therefore submit duplicated work.** Failure to meet these requirements leads to a violation of the academic integrity principles as stated in your syllabus.

Grading Criteria: See the grading criteria information in your course **syllabus addendum** (Click Home and look towards the middle of the page).

Objective: Demonstrate your understanding of using ArrayLists to process data in Java. This assignment is based on the material covered in **chapter 11** of your textbook.

Deliverable: Use the file format:

- **firstNameLastNamecit130_Pa11.java** for the source file of your class. You should only submit one Java source file. There is no need for creating a utility class for this exercise.

Documentation: See your course **syllabus addendum** file in Canvas for the required heading documentation and other notes. Make sure you have the header section properly created using tags such as `@author`, `@file`, etc. **Methods need to be properly documented.** See the syllabus addendum file, part 2 (e) on proper process to document each method.

Assignment: Write a program to practice with the use of ArrayList in Java.

Constraints:

- Do not use break/continue in your loops.
- Do not use the System.exit() method to exit the program.
- Do not use global variables. Parameters must be passed to methods. The only global variable allowed would be the definition of the Scanner class, or in very specific situations as strictly required by the program requirements.

Process:

Write a program that uses the **ArrayList** class to store data for a purchase processing system.

- a) Define a **named constant** for the unit cost of all items and set it to \$20.
- b) Prompt the user to enter the number of purchases made at a business. Validate the number of purchases to be a value between 1 and 10.

- c) Prompt to user to enter full name (i.e., foo bar) of a person making a purchase,
- d) Prompt to user to enter the number of items a user wishes to purchase,
- e) Create a new arraylist to hold the total cost of items purchased by a user.
- f) Display the message "PURCHASE REPORT".
- g) Display the data using the **toString** method for each arrayList.
- h) Ask the user to enter the name of a person and check if the person exists in the arraylist.
Your search must be case insensitive.
- i) Display the message "**USER** found" (where USER is replaced by the user's name), user's corresponding number of items purchased, the unit cost and the total cost. Otherwise, display "**USER NOT** found" where **USER** is replaced by the user's name.
Create a **continuous loop** that keeps asking the user for more names to search quitting when a sentinel value has been entered.
- j) Ask the user for the number of data to add and allow the user to add more names.
- k) Ask the user for the name of a person to remove and allow the user to remove the data for this user. If the name to be removed cannot be found, display an error message.
- l) Display the message "FINAL PURCHASE REPORT".
- m) Display the size of the arraylists using the size() method. Of course, they run parallel, so they would have all the same size. Make sure to be very descriptive in your output.
- n) Display the data in a readable tabular format.

Sample Interaction:

```

Enter the number of purchases: -1
Invalid entry - try again: 12
Invalid entry - try again: 2
Enter the person's full name: foo bar
Enter the Number of items purchased: 3
Enter the person's full name: candy cane
Enter the Number of items purchased: 10
PURCHASE REPORT
[foo bar, candy cane]
[3, 10]
[60.0, 200.0]
Who would you like to search for - enter exit to end search? foo bar
foo bar found
3 20.0 60.0
Who would you like to search for - enter exit to end search? harry cane
harry cane NOT found
Who would you like to search for - enter exit to end search? exit
Enter number of data to add: 2
Enter the person's full name: shark tank
Enter the Number of items purchased: 4
Enter the person's full name: far out
Enter the Number of items purchased: 70
Who would you like to remove? far out
The size of the array lists is: 3

```

Name: foo bar - Number of items: 3 - Total cost: \$60.0

Name: candy cane - Number of items: 10 - Total cost: \$200.0

Name: shark tank - Number of items: 4 - Total cost: \$80.0