Lab 3
125 Points
SalesTracker
Readings 7-9
GitHub Classroom Clone
Project Name

In our third lab of the semester we will create a **Java CLI application** that enables a manager to enter **up to 20** salespeople's names, IDs and annual sales. Each salesperson must be an object and must be part of one array. **(No parallel arrays permitted.)**

The manager can choose to add a new salesperson until there are 20 total. Once 20 salespeople are present, no new data can be added.

No duplicate IDs should be entered. Once the manager is finished, display all the salesperson data in an easy to read format.

In addition, the manager can change a salesperson's annual sales by entering a salesperson's ID. Make sure to check if that **salesperson exists and inform the manager**. After the change, re-display the list.

Please note: IDs are **eight digits long** and cannot be duplicated.

Lab Parameters

We need to use two separate exercises for our lab. To create the object class use the first half of **Exercise 6A on pages 433-434** of the textbook. **Make sure to add the salesperson's name.** You only need the information on constructing the **Salesperson** class not the **DemoSalesperson** application in this exercise.

After creating the **Salesperson** class, use **Exercise 5 on page 485** of the textbook to create your application class. All parameters and logic in this section should be included, **except for deleting the record**. **Make sure to allow for input of the salesperson's name as well.**

Please note the following important distinctions:

- There is no actual database for this lab even though the exercises says to "pretend" it's a database. We are working with arrays and no data will be stored between application runs.

CreateSalesperson.java

Your main application class. The logic noted in **Exercise 5 (Chapter 9)** should be included here except **there are no deletions**.

Remember that the application class controls the program’s logic.

Salesperson.java

Your class used to create each of the 20 potential salesperson objects. Each contains a name, ID, and annual sales. This is from Exercise **6A (Chapter 8)**.

Make sure to add ASCII design and include your company’s name and other important information in the output.

Challenge Parameters

For those who want to create a more robust application, consider the following:

1. **[Moderate Challenge]** Display the list **sorted by ID** (ascending).
2. **[Somewhat Difficult Challenge]** Allow the manager to choose which item should be sorted and in what order (ascending versus descending) before displaying the list.
3. **[Difficult Challenge]** Do not populate an entire 20-item array. Instead use a dynamic array of objects. Allow the manager to add, edit, **delete**, and sort all of the salesperson objects. Choosing this option requires you to code a large portion of the lab in a different manner than discussed in class and should only be attempted if this is not your first programming class. **This option must include all features noted in #1 and #2.**

It is better to have a completely coded, documented, and functional program without challenge parameters than it is to have a non-working or partially working lab with challenge parameters.

As a reminder: The tutor is instructed to not help with any challenges.

Lab Parameter Reminders

- ID must be eight characters and not a duplicate.
- No parallel arrays.
- Each salesperson must be an object.
- Only 20 entries permitted.
- Do not delete the record.

Important items to note

- Clone your Lab template from the GitHub classroom URL in eLearning
- Make sure to commit with a detailed message and push your final code at a minimum. ***Multiple commits and messages at major logic completions are preferred.***
- Use NetBeans for this lab
- Follow variable and file naming conventions
- Use appropriate data types
- Document every file that you create and/or change. You must explain the new concepts and approaches (e.g., classes, etc.) in great detail. Other items such as **System.out.println()** are older concepts and do not need to be documented in detail but should be noted.
- Include your classID as appropriate

Deliverables

This lab should be turned in as follows:

- A completed final commit with descriptive commit message pushed to your lab GitHub repository.
- Your GitHub repository URL in the eLearning dropbox comment section.
- A screen shot of your final GitHub commit uploaded into eLearning named **Lab3classID.png.** So Bubba Jones would take a screenshot of lab3-bubbacoder and name it:

Lab3bjones4242.png

Refer to the **Documentation Guide** at for guidance on comments and lab preparation. Make sure to watch the GitHub process video if you have commit issues.

When you are finished, make sure save, commit, and push to your GitHub repository
AND
submit your image and repository URL into the eLearning DropBox.