Lab 2
100 Points
Sell-O-Tron
Readings 3-6
GitHub Classroom Clone
Project Name

In our second lab of the semester we will create a **Java CLI application** that enables a store owner to enter sales information in order to produce a detailed invoice.

The store owner will enter the invoice number, name of the item, and amount of sale. In turn, the application will create the invoice number, check the item name, calculate sales tax and output the invoice with all pertinent information to include the final bill.

Lab Parameters

Using **Exercise 14A and 14B on page 350** of the textbook as a starting point, create an application with the following:

- **Purchase.java** (class used to create objects)
- **CreatePurchase.java** (application class)

Purchase.java

The object class should contain the appropriate get/set methods as discussed in the exercise, but make sure to include the item's name as well in a get/set method. Remember to follow encapsulation and data hiding approaches to restrict access.

CreatePurchase.java

The application class should contain the appropriate parameters as discussed in the exercise. However it should also include a means to limit item names to at **least five different specific item names**.

Note: One of the major decisions with this lab is deciding what should go in the application class versus the object class. Make sure to explain your approaches in the documentation.

Once the application has all the input and processed it, output the results with all relevant information to create a detailed invoice. Make sure to add ASCII design and include your company's name and address.

Challenge Parameters

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

1. **[Moderate Challenge]** Allow for more than one invoice to be created without starting the program again.
2. **[Difficult Challenge]** Allow for more than one item on a single invoice.
3. Add additional range checks in the application or class as appropriate.

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.

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

Remember, everything we learned up through this lab is important.

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 **Lab2classID.png**. So Bubba Jones would take a screenshot of lab1-bubbacoder and name it:

Lab2bjones4242.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.