## Name Gilberto Sandate Sebastian Levario

## Date April 20, 2024

CS 3331 – Advanced Object-Oriented Programming – Term Spring 2024

## Instructor Bhanukiran Gurijala

## Assignment

This work was done individually/as a team and completely on my/our own. I/we did not share, reproduce, or alter any part of this assignment for any purpose. I/we did not share code, upload this assignment online in any form, or view/received/modified code written from anyone else. All deliverables were produced entirely on my/our own. This assignment is part of an academic course at The University of Texas at El Paso and a grade will be assigned for the work I/we produced.

# **Program Explanation**

In this section, explain the overview of the assignment.

What did you do?

How did you tackle the problem?

What techniques did you use to solve the problem?

Did you break the problem into smaller problems? Explain.

This section my team and I went over our code and made few changes. At first we were fixing our past mistakes. Then, when we were implementing more and more methods, such as the new admin and user functionalities, we made bigger, more impactful changes. For example, my peer thought that it was better to separate our main file from the little methods, so that was done. We had to refactor our code more this time in order to accommodate all the new additions. We also used try-catch statements. We really didn’t break down the problems into smaller problems, it was more of a “do-a-little bit-at-a-time-between-all-of-us” type of thing. Like it was split between all of us.

# **What did I learn?**

What did you learn as a result of this assignment?

How can my solution be improved?

What ideas do you have about another way to solve the problem?

How long did it take me to complete this lab assignment?

As a result of this assignment, we learned how HashMaps really work in practice. We also learned how to use GitHub as a team in order to work on different problems at the same time without having to wait for each other to finish with their part. I really can not think of a good way that our solution can be improved as the solution implemented seems to be best, but I could be wrong. Another way to solve this problem could be to use files or arrays to store the cars’ and users’ information, but I think that doing so would lead to the time complexities to reach a higher value. This lab assignment took us about a week working on it on and off.

# **Solution Design**

What did you do in this program?

What was your approach to solving this problem?

What data structures did you use? Why?

What assumptions, if any, did you make?

In this program we built on what we previously had on part one and added to it. We gave the admin class functionalities. We made it so admin could add and/remove cars. They are now also able to view a car’s revenue. We also added more functionalities to users. We made it so they could return a car if they regretted their purchase. We also added discounts and taxes to the prices of cars, which were different depending on the users’ membership status. We used HashMaps in order to achieve this because we believed that they would work best in these scenarios. HashMaps working like Python’s dictionaries with keys made them best. The assumptions that we made were that discounts were applied before the taxes were. We also assumed that when a car was removed, if would be removed completely from the csv file.

# **Testing**

How did you test the program?

Did you use black-box, white-box testing, or both? Why?

Did you test the solution enough? How can the testing practices be improved?

What are the test cases I used?

Did you break the program and use that to improve it?

We tested the program by actually running the runShop file, which is the file with the main method. We mostly used black-box testing to test our program, meaning we only looked at what went in and came out. We also used white-box testing whenever we encountered a problem or error that didn’t make much sense to us. I think that we did test the solution enough as by the end our program was fine. We also did have to “break” the program to improve it. We split up the main file into two files, one that had the main method, which had what the user would see, and one with all the methods that the main file was using.

# **Test results**

Describe the results of your tests.

Include any console outputs showing your results.

Include any text document results of your tests.

The results of our tests varied a lot at the beginning. There was a point in our testing were the “Display all Cars” was not working correctly and it would print all the cars as if they had no attributes. We were scared of pushing to Git and after about 10 minutes we figured out that the cause of this was. We found out that the file that was getting read was not the car file, but instead the user file. After that change everything was smooth sailing more or less.

# **Code Review**

Explain how you conducted a review of your code. Describe how you checked each part of the code review checklist.

We literally went through our code, like each of us on our own. And whenever one of us saw that there was something either missing or wrong, then we would all look into it and work on it. We would go into the checklist so often that most of the time it was always open and ready. For instance, there was this one time when there was a mismatch between us where we all understood the “Remove Car” functionality. I had thought that it meant to fully remove it from the file as if it had never been there, while my peers thought that it meant to just remove one from the availability. After a bit of discussion and back and forth we ended up going with removing it completely from the csv file.