Names: Carlos Cabral & Edgar Rodriguez
Date: April 9, 2024
CS 3331 – Advanced Object-Oriented Programming – Term Spring 2024
Instructor: Dr. Bhanukiran Gurijala
Assignment: Project 1 - Car Dealership

This work was done 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.

1. **Program Explanation**
    - In this section, explain the overview of the assignment.
    - What did you do?
        - We (Carlos and Edgar) worked togethers as a team to create the software for a car dealership.. Then we proceeded to write the code.
    - How did you tackle the problem?
    - What techniques did you use to solve the problem?
        - We defined the use cases, and classes with UML diagrams.
        - We created a couple of sets of abstract classes for the
            - Person>Customer
            - Car > CarTypes
    - Did you break the problem into smaller problems? Explain.
        - Yes, we separated the code into different classes based on functionality (implementing the Single-Responsibility Principle)

2. **What did I learn?**
    - What did you learn as a result of this assignment?
        - We learned putting together the covered principles acquired throughout the class. Collaborating as a team, coordinating tasks, and making progress collectively presented a significant learning opportunity, specifically not being able to start a new task that is dependent on another task.
    - How can my solution be improved?
        - We hardcoded the name of the file, it would be nice to receive files without hardcoding the name in the code. Also, we implemented formatting to the log file, but additional formatting could improve legibility
    - What ideas do you have about another way to solve the problem?
    - How long did it take me to complete this lab assignment?
        - 3 weeks. It took us more than expected.

3. **Solution Design**
   Wrote a runShop Class that contains the main method. fileReader reads and writes. We created an abstract class for Car and extended it into different car

types. Similarly, we created an abstract Person class and we extended it into a Customer class. We configured a logger that outputs a log.txt file.
For data structures, we used two dimensional arrays and string arrays.

4. **Testing**
    - How did you test the program?
        - We did manual/conventional testing. But we also defined test cases.
    - Did you use black-box, white-box testing, or both? Why?
        - White box, since we were aware of the logic and we could identify what could go wrong.
    - Did you test the solution enough? How can the testing practices be improved?
        - Our testing could be improved by sharing the program with other users and see how they interact with it without knowing how it works.
    - What are the test cases I used?
        - Wrong login information
        - Files not found to read
        - Wrong input to menu (not an integer)
    - Did you break the program and use that to improve it?
        - Yes. We encountered some issues and improved them.

5. **Test results**
    - Include any console outputs showing your results.
        - Example of wrong login information:
            - Apr 09, 2024 11:46:41 PM runShop main
            - INFO: User attempted login
            - Invalid login
            - incorrect login information
            - Apr 09, 2024 11:46:41 PM runShop main
            - INFO: Application Ended
    - Include any text document results of your tests.
        - Please see “log.txt” that includes the log of errors.

6. **Code Review**

- We checked the code covered the use cases defined on the requirements. We also checked bugs and logical errors. Identified the lack of try/catch exceptions. Shared input on code formatting style.