## Alejandro Olivas

## Date 4/4/2023

CS 3331 – Advanced Object-Oriented Programming – Spring 2023

## Dr. Daniel Mejia

AOOP PA4

This work was done individually and completely on my own. I did not share, reproduce, or alter any part of this assignment for any purpose. I 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 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 produced.

NOTE: **Please write in complete sentences (paragraph form).** Check grammar, punctuation, and ensure your writing is clear. Write enough to make sure you clearly explain each part of the question.

# **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.

For this project the main parts were to allow a user to search by airport code, allow an employee to view airport information, allow an employee to run an automatic purchasing feature, and to allow and employee to print an electronic ticket summary of a user’s tickets. I was able to do everything successfully mainly because I split the tasks into different methods/files. For the first two requirements I was able to create a Hash Map of airports that you can search by their code which made it a lot easier to split up. For the automation requirement I used a lot of the pre-existing methods while purchasing tickets but didn’t print the outputs. For the last requirement I split into another file to be able to write information a lot easier and split up all the users’ tickets.

# **What did I learn?**

What did you learn as a result of this assignment?

How can my solution be improved?

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

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

The main thing I learned from this assignment is to always try to make your code flexible so it can be easier to change or easier to add extra features in the future. The main way I think my solution can be improved is by rewriting the purchase tickets method to either do an automatic ticket purchasing or the regular ticket purchasing for the user with all the different prints. I didn’t go about solving it this way because I feel like the two functionalities are different enough to be split into their own methods and it can introduce more room for error like if something is suppose to print to the user but doesn’t. To finish this whole lab I think it took me a total of 4 hours mainly because the airport class was set up in PA3 to be easily changed for the future so I didn’t have a lot of issues when it came to adjusting it.

# **Solution Design**

What did I do in this program?

What was my approach to solving this problem?

What data structures did I use? Why?

What assumptions, if any, did I make?

To solve the first two problems I created a Hash Map with all the airports in the Flight Factory and then return all the Hash Map to the data file to be able to use it later. In the airport object I also gave it a set that has all the Flight ID’s that the airport is the Origin for and a separate list of Flight ID’s that the airport is the destination for. Using those two lists I’m able to take the Origin code and destination code that the user gave and print all the flights that have common origin and destination ID’s. To print the airport information I would simply look up the code given by the employee in the Hash Map and run a method in the airport object that printed all the information it asks. For the Ticket automation I would read each line in the file, get all the needed information, and because I already have the methods that purchase the actual ticket, I simply called those methods in the Data class like if it was running in the regular ticket purchasing steps.

For the electronic ticket I would loop through every ticket the user has, pass the ticket and the flight objects to a “WriteElectronicTicket” file, and then that file will create/append all the information required to the csv on ticket at a time. I did ask if the header of the automation will always be the same, so I hard coded the cols of the file, but this can easily be adjusted to work like it does for the Flight/Customer file.

# **Testing**

How did I test my program?

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

Did I test my solution enough? How can my testing practices be improved?

What are the test cases I used?

Did I break my program and use that to improve it?

I mainly used Black Box testing to be able to test my program when it was finally written, and I tried to input every possible way a user can mess up. While writing my code I did use a lot of print statements, that were later deleted, and a testing file to test individual parts of my code to verify that they were doing the task that I intended. There possibly is a test case that I might have missed but I tried to think of every possible entry that a user can input. I wasn’t able to break my program during the testing. We were also allowed to let a class mate run our code during class and try to break it. When this test was ran my class mate was also unable to break my code or got promoted to try again when an invalid value was given.

# **Test results**

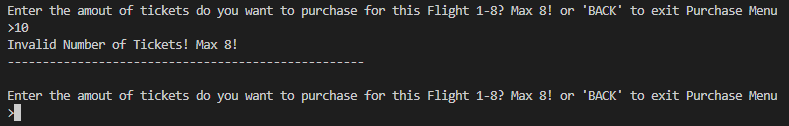
Describe the results of your tests.

Include any console outputs showing your results.

Include any text document results of your tests.

The results of my tests were all good because I did put multiple switch statements that if it doesn’t receive an expected input then it will ask the user to keep trying. I put a few examples of some errors that you can get with the new functionality.

Trying to Purchase above 8 tickets:



Inputting a First/Last name that doesn’t match the username’s First/Last name:

Text

Description automatically generated

Trying to purchase a flight that has been canceled:

Text

Description automatically generated

# **Code Review**

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

The major thing I did to do a code review was to do it on a day other than when I last looked at my code, allowing me to approach it objectively. I then opened the professor's Code Review Check List and began checking each item off to make sure it fits the requirements. The logic in the code is all implemented as planned, and it all makes sense to me. I was able to find an error in the automation result while conducting my code review and notified the team about the error. Overall, I wouldn’t change much from what I did except to make it easier to read.