## Daniel Guevara

## October 4, 2022

CS 3331 – Advanced Object-Oriented Programming – Fall 2022

## Dr. Mejia

## Programming Assignment 2,3.

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.

# **Program Explanation**

PA2:

Programming assignment two was a direct continuation of programming assignment one, students were tasked with extending their UML class diagram and their UML use case diagram to match the new requirements for this lab. Those requirements included updating the user interaction functionality to include the role of an administrator. Administrator interaction meant the program would be able to handle providing all the information about an event whenever inquired. Inquiries for events were to be handled through either the event’s name or id. All user interaction was also to be written into a log file upon termination of the program. Additionally, upon termination of the program, a new event list should have been written containing all the new information of the events.

To complete this task, a new UML Diagram was completed basically starting from scratch. Instead of completely building off PA1, the foundation of this assignment was different. This decision was made to fulfill the functionalities required. Unlike my previous assignment, where I tried to do everything all at once, I decided to divide things into different tasks. Hence, I created different methods with different functionalities. For instance, instead of using a switch case inside of the main method to get venue/ticket/event information depending on user input, I created a method that would return the capacity/price of a ticket, with the ticket tier being the parameter. Another way I did this was by adding additional methods to the customer class, these methods included returning whether or not a user had a sufficient balance to purchase tickets. Also, another class called account was created which included methods that withdrew and deposited money into a user’s account.

Finally, inside of the main class, methods that handles logging in, purchasing tickets, inquiring an event through id, inquiring an event through name, and writing a new event.csv were created. This all made the functionality of the main method much clearer and had a much more object-oriented design.

PA3:

Students were tasked with extending their UML class diagram and their UML use case diagram to match the new requirements for this lab. This include adding a few methods to the class diagram and adding a few functionalities to the use case diagram. After completing this part, students needed to implement a dynamic file reader, meaning it could handle reading a file whose columns may be in random order. After completing this, students needed to add a few functionalities to their program, which included adding a 10% discount to ticket miner members and also adding a Texas sales tax to all sales/purchases. Lastly, each event should be able to keep track of total tax revenue generated as well as total discounts offered, adding onto that, the program was to keep track of all user savings.

# **What did I learn**

PA2:

Upon completion of this assignment, I learned the importance of creating a UML diagram before writing code. When I begin assignments, I like to just jump into coding rather than organize. This time I did it differently and made sure to keep everything clean and clear. I believe there was a better way of structuring my code, for instance some attributes may have belonged to other classes, however at the time I feel that I structured it to the best of my understanding.

It’s hard to put into hours directly how long this lab assignment took me. What I can say is it took more than 10 days of work. There were some days where I worked on it for maybe an hour or two, but then there were some days where I started during the day, and I was still working on it at sundown.

PA3:

I learned to thank my past self, thankfully I set up a proper code structure/foundation in the previous programming assignment. Having said this, I only needed to add a few attributes to my event and customer class which kept track of tax revenue and total savings for both event and customer.

# **Solution Design**

PA2:

I completed this program using clear object-oriented programming while also implementing a factory design pattern. Not to be corny, but as I was writing the words, “public abstract class” and while writing the word, abstract, the word abstraction came to me. Suddenly CS2 memories came back to me, that’s when I decided to take a piece of paper out and break this assignment into as many little pieces as possible.

Array List was my data structure of choice, I used them to hold all the information from the customers csv along with the information from the event list csv. I like the idea of appending to a data structure as many times as we need to and not having to worry about something being a fixed size.

PA3:

I added a savings attribute to the customer class which was responsible for holding the total number of savings for each customer. I added a method that added savings to this attribute and called it in the main method whenever a ticket miner member was purchasing a ticket. Next, I added an attribute to the event class that was responsible for holding total amount of discounts generated by an event, using the same idea from the customer class, I created a method that added to this attribute. Then I called this method whenever tickets to an event were purchased. To keep track of total tax revenue generated, I simply added the total revenue generated by an event and applied a Texas sales tax to it.

# **Testing**

PA2:

To test the functionality of my program I incorporated white box testing. First, I tested to see if my program can stand on two legs and just run without issue, next I tested to see if the event information would update upon the purchase of tickets, next I checked to see if inquiring about an event worked correctly for both option a, which was through event id and option b, which was through event name. After that I tested to see that the inquiry option was only available to users who said they were administrators. Upon testing at first, I received many, many errors, but after that I was able to trace my issues and solve them.

# **Test results**

Text

Description automatically generated

The above picture was testing to see an inquiry through an event id, the test was successful!

Text

Description automatically generated

The above picture was testing to see an inquiry through an event name, the test was successful!

Text

Description automatically generated

The above picture was testing to see if a non-administrator had the option of inquiring about an event, the option was not available making the test successful!

PA3:

To test my new implementations, simple white box testing was used to ensure each new implementation worked as intended.

Picture below proves each event’s tax revenue generated as well as all the discounts used.

Text

Description automatically generated

The picture below proves each user’s total savings are being saved.

Text

Description automatically generated

# **Code Review**

PA2:

A code review was conducted of this lab assignment by first reading the entirety of the code checklist provided to us. After that, I went one by one for every bulletin point in the document to be sure I reached the necessary requirements needed from the checklist.

PA3:

A code review was conducted by going line by line for all new lines of code added, ensuring that it meets the requirements of the code review checklist provided to us.

**Reason for lateness:**

**PA3:**

Reason for my lateness is I have trouble managing my time wisely between work and school, I know I need to be more proactive with these programming assignments and I promise that I will.