The University of Texas at El Paso
Department of Computer Science
CS 3331 – Advanced Object-Oriented Programming
Instructor: Daniel Mejia
Fall 2023

Programming Assignment 2

Academic Integrity Statement:
This work is to be done individually. It is not permitted to share, reproduce, or alter any part of this assignment for any purpose. Students are not permitted from sharing code, uploading this assignment online in any form, viewing, receiving, or modifying code written from anyone else. 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 produced individually by the student.

Objective:
Utilize Object-Oriented programming design and principles to create a system.

Instructions:
Your code must be written in Java. You must submit your assignment through GitHub Classroom. In the comment heading of your source code, you should write your name, date, course, instructor, programming assignment 2, lab description, and honesty statement. The honesty statement must state that you completed this work entirely on your own without any outside sources including peers, experts, online sources, or the like. Only assistance from the instructor, TA, or IA will be permitted.

Scenario:
You have recently been hired to work for *TicketMiner*, a company that sells tickets for sporting events, concerts, festival events, etc. You have a few customers that are interested in creating their events using your system.

Part A:
Read the requirements described in Part B to complete Part A. Part A must be completed before implementing the requirements in Part B

1. Update the UML Class Diagram to structure your code using the classes, requirements, and concepts described in Part B
    a. Update the UML Class Diagram from PA1
2. Update the UML Use Case Diagram (Level II) for your system
    a. Update the UML Use Case Diagram from PA1
    b. 2 actors
    c. 4 Use Cases

Part B

Before you start:

Verify that you are following Java coding standards/conventions. Please use this Google Styling guide to help: https://google.github.io/styleguide/javaguide.html. Ensure that you are writing code is modular and object oriented. Please utilize the comments provided to you in previous PA assignments.

1. Verify that all functionality from PA1 (and before) is fully implemented and complete
    a. You must complete all functionality from the previous assignment
    b. Each assignment builds upon the previous assignment
2. Update your system to include a design pattern
    a. You may need to refactor your code
3. Update user interaction functionality
    a. Ask if the user is a customer
        i. If the user is a customer, ensure that all customer functionality works
    b. Ask if the user is a system administrator (no log in necessary)
        i. Administrators have access to inquire about any event
            1. Example:

Console:	"A. Inquire event by ID"
	"B. Inquire event by name"
User:	"A"
Console:	"What is the ID of the event?"
User:	10
Console:	========== Event ID: 10 UTEP Football 10

11/6/23
7:05 PM
Event Type: Sport
Event Capacity: 58000
Total Seats sold: 0
Total VIP Seats Sold: 0
Total Gold Seats Sold: 0
Total Silver Seats Sold: 0
Total Bronze Seats Sold: 0
Total General Adm Seats Sold: 0
Total revenue for VIP tickets: $0.00
Total revenue for Gold tickets: $0.00
Total revenue for Silver tickets: $0.00
Total revenue for Bronze tickets: $0.00
Total revenue for General Admission tickets: $0.00
Total revenue for all tickets: $0.00
Expected profit (Sell Out): $459940.00
Actual profit: $-681500.00
==========

2. Example:

Console:	"A. Inquire event by ID"
	"B. Inquire event by name"
User:	"B"
Console:	"What is the name of the event?"
User:	"UTEP Football 1"
Console:	========== Event ID: 1 UTEP Football 1 9/4/23 7:05 PM Event Type: Sport Event Capacity: 58000


```
Total Seats sold: 0
Total VIP Seats Sold: 0
Total Gold Seats Sold: 0
Total Silver Seats Sold: 0
Total Bronze Seats Sold: 0
Total General Adm Seats Sold: 0
Total revenue for VIP tickets: $0.00
Total revenue for Gold tickets: $0.00
Total revenue for Silver tickets: $0.00
Total revenue for Bronze tickets: $0.00
Total revenue for General Admission tickets: $0.00
Total revenue for all tickets: $0.00
Expected profit (Sell Out): $352930.00
Actual profit: $-681500.00
====================
```

ii. Print a summary of all events and their information

4. Your system should be able to handle the following
    a. Inquire all information of event
    b. Inquire the number of total seats remaining
    c. Inquire the number of VIP seats remaining
    d. Inquire the number of Gold seats remaining
    e. Inquire the number of Silver seats remaining
    f. Inquire the number of Bronze seats remaining
    g. Inquire the number of General Admission seats remaining
    h. Inquire the number of seats remaining in total (Excluding reserved)
    i. Compute the total amount collected from VIP seats sold
    j. Compute the total amount collected from Gold seats sold
    k. Compute the total amount collected from Silver seats sold
    l. Compute the total amount collected from Bronze seats sold
    m. Compute the total amount collected from General Admission seats sold
    n. Compute the total amount collected from all seats sold
    o. Compute the total profit expected for the event (sold out)
    p. Compute the current profit (may be negative)
5. Log each action that is taken
    a. Write to a log file (this file can be written at the termination of the program)

6. Upon the termination of the program (i.e., by user inputting “EXIT” while in the main menu only)
    a. Write a new event list (new csv file – do not overwrite the original input) with the updated values. Note: You will have the new columns not part of the original, including:
        i. VIP Seats Sold
        ii. Gold Seats Sold
        iii. Silver Seats Sold
        iv. Bronze Seats Sold
        v. General Admission Seats Sold
        vi. Total VIP Revenue
        vii. Total Gold Revenue
        viii. Total Silver Revenue
        ix. Total Bronze Revenue
        x. Total General Admission Revenue
    b. Write a new customer list (new csv file – do not overwrite the original) with the updated values (tickets purchased, transaction count, etc.)
7. Handle all exceptions appropriately
8. Write the Javadoc for your system
9. Write a lab report describing your work (template provided)
    a. Any assumptions made should be precisely commented in the source code and described in the lab report
    b. Lab report should contain sample screenshots of the program being run in different circumstances including successful and failing changes
10. Complete an individual code review on your code (template provided)
11. Schedule a demo with the TA/IA
    a. Be prepared to provide a detailed demo your system
12. **If submission is past the deadline** Your report must have an additional section entitled “Why I submitted late”. In that section, explain the reason why your submission was late. (Note: you will still be penalized the typical late penalty)

Deadlines:

October 3, 2023, by 11:59pm (Current Progress Commit) – GitHub classroom:

1. UML Class Diagram Progress (.pdf)
2. UML Use Case Diagram Progress (.pdf)
3. Current Progress Source Code (.java) – Commit current progress up to this point

For each item (1-3)

a. Does not have to be complete
b. Should be a significant amount of work done (as determined by instructional team)
c. TA/IA will review for progress only

<u>October 8, 2023, by 11:59pm - GitHub Classroom:</u>

1. UML Class Diagram (.pdf)
2. UML Use Case Diagram (.pdf)
3. Source code (.java files)
4. Lab report (.pdf file)
5. Javadoc (entire doc folder)
6. Updated Event Sheet (.csv)
7. Updated Customer Sheet (.csv)
8. Log (.txt)

