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

Programming Assignment 4

Academic Integrity Statement:
This work is to be done in teams of 4 people. 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 4, 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, special 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 should be completed before Part B

1. Write a refactored UML Class Diagram to structure your code using the classes, requirements, and concepts described in Part B
2. Write a level II UML Use Case Diagram based on Part B
    a. 3 Use Cases
    b. 2 Actors
    c. 1 extend
    d. 2 include
3. Write a UML State Diagram describing purchasing event ticket(s) as an individual customer

Part B

1. Conduct a code review on your teammates code
    a. Understand what they did
    b. Understand their approach
    c. Ask questions
    d. Give constructive feedback
    e. Discover and communicate the best way to incorporate their code
2. Combine code from all teammates into a single program
    a. Do not simply use one person's entire code
    b. Using Javadoc – write where each piece of code came from (i.e. Taken from Bob, Taken from Alice, Taken from Samantha)
3. Refactor your existing code
    a. Your code should handle all functionality from Programming Assignment 3 (PA3)
    b. Fix anything that should be corrected
        i. Appropriate data structures
        ii. Appropriate use of objects
        iii. Relationships between objects
        iv. Algorithms and complexity
4. Add new event manager functionality
    a. Create new events from the console
        i. Events do not necessarily need all fields. Required fields are:
            1. Event ID (auto generated by system)
                a. largestEventID++
                b. Do not use size of data structure (holding all the events) + 1, this may not necessarily be the next ID number
            2. Event Name
                a. Any String
            3. Event Date

a. MM/DD/YYYY
4. Event Time
    a. XX:XX AM (or PM)
5. Venue (use only the venues already determined in venue list)
    a. Populate and choose from a selection on the console
        i. Sun Bowl Stadium
        ii. Don Haskins Center
        iii. Magoffin Auditorium
        iv. San Jacinto Plaza
        v. Centennial Plaza
    b. Number of seats for each ticket level can be programmatically computed and determined by venue (these percentages come from input file)
        i. VIP – 5% of venue capacity
        ii. Gold – 10% of venue capacity
        iii. Silver – 15% of venue capacity
        iv. Bronze– 20% of venue capacity
        v. General Admit – 45% of venue capacity
        vi. Reserved Extra – 5% of venue capacity
6. General Admission Price (max General Admission price $4000)
    a. Prices for remaining levels can be programmatically computed and determined
        i. VIP – 5 times more than GA
        ii. Gold – 3 times more than GA
        iii. Silver – 2.5 times more than GA
        iv. Bronze– 1.5 times more than GA

b. Include all event print functionality from (PA3)
c. Include if the event will have fireworks in event information console print
d. Include the total cost (Venue Cost + fireworks) in event information console print

5. Automatic Purchasing Functionality
    a. Ticket Buyer Automation
    b. Use the “Autopurchase.csv” file (or the different versions of this file)
        i. There are multiple versions, all should work (the only difference is the number of purchases)
    c. You should still follow the constraints of purchasing a ticket manually
        i. Do not sell if there is not enough money in the account
        ii. All successful purchases should give a customer an “invoice”

iii. All successful purchases should be maintained in each individual event

6. User Interaction
    a. Customer interaction Functionality (PA3)
    b. Add Event Manager Functionality
        i. You may extend the already established functionality to have additional options
        ii. Ability to write an "Electronic Invoice Summary" (.txt file) for a specific customer that contains:
            1. Event Type
            2. Event Name
            3. Event Date
            4. Ticket Type
            5. Number of Tickets
            6. Total Price
            7. Confirmation Number
        iii. "Electronic Invoice Summary" should list all ticket purchases made by the customer
7. Terminate the program by using the writing "EXIT" while in the "main menu"
    a. Write a new event list (new csv file – do not overwrite the original input) with the updated values (seat information, money earned, etc.)
    b. Write a new customer list (new csv file – do not overwrite the original) with the updated values (tickets purchased, transaction count, etc.)
8. Create an Electronic Invoice Summary for six customers (each team member, and two of your favorite Disney characters on the list)
    a. Six different .txt files (one for each)
9. Write test cases and a test suit to test at least two code methods. To write these test cases, you can use the Junit package.
    a. Ensure you have JUnit added to your project.
    b. Create separated files for each test case and test suit.
10. Handle all exceptions appropriately
11. Write the JavaDoc for your system
12. 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
    c. Write an additional section describing the demo of your partner
        i. What questions did you have about your partners functionality?
        ii. What concerns do you have about your partners functionality?
        iii. How did you try to break it?

        1. What test cases did you use?
    d. Explain why and how this is a black or white box texting
13. Complete an individual code review on your code (template provided)
14. Demo with another team, only functionality – don't share code
    a. Make sure that they understand what you're talking about
    b. Make sure that they cannot break your code
    c. Partners should focus on ensuring all requirements are met (functionality)
    d. Partners should try to break the system
15. Schedule a demo with the TA/IA
16. **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 31, 2023, by 11:59pm (Current Progress Commit) – GitHub classroom:

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

For each item (1-4)

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

November 5, 2023, by 11:59pm - GitHub Classroom:

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

