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

Programming Assignment 1

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 1, 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 must be completed before implementing the requirements in Part B

1. Write a UML Class Diagram to structure your code using the classes, requirements, and concepts described in Part B
2. Write a UML Use Case Diagram (Level I) for your system (Note: This diagram will not have the same requirements as separately assigned homework assigned, but you may use that assignment to help you begin this portion of the work)
    a. 2 actors
    b. 3 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. Create the following classes (Note: some may be abstract)
    a. Customer
    b. Concert
    c. Event
    d. Festival
    e. Sport
    f. Arena
    g. Auditorium
    h. OpenAir
    i. Stadium
    j. Venue
    k. Invoice
    l. RunTicket (Main Method)
    m. Any other classes that you believe will be beneficial to help with successfully implementing this program
2. Read files with information and store the information appropriately
    a. Pick a data structure that is appropriate
        i. Consider the time complexity
        ii. Consider space complexity
    b. Consider use of objects
3. Allow for user interaction
    a. Ask for first & last name to log in
    b. Allow the person to log in using username and password
    c. Allow the person to purchase a ticket for the event (1-6 tickets per transaction)

i. Buy tickets to only one event per transaction
    1. Tell user the price for each ticket level
ii. Buy only one ticket type per transaction
iii. Verify that there are still tickets available
iv. Provide the customer with a "invoice" in which it has the quantity of tickets, total price for the purchase, and a confirmation number
    1. One "invoice" per transaction (regardless of the quantity)
v. User will have an "invoice"
vi. Event will have a list of tickets purchased for the event
vii. Include reasonable prompts to make the transaction understandable

d. Customer should only be able to purchase tickets if they have enough money
e. Update the customers available money after their ticket purchase

4. Log each action that is taken
    a. Write to a log file (this file can be written at the termination of the program)
5. Terminate the program by typing "EXIT" while in the main menu only
6. Handle all exceptions appropriately
7. 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
8. Complete an individual code review on your code (template provided)
9. Schedule a demo with the TA/IA
    a. Be prepared to quickly demo your system
    b. Be prepared to discuss your UML Class and Use Case Diagrams
10. **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:

September 19, 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>September 24, 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. Log (.txt)

