



PROJECT

COURSE : DATA STRUCTURES

COURSE CODE : SWC3344/SWC3064

SUBMISSION DATE : 6TH NOVEMBER 2023

1. This assignment aims to fulfil CLO – Develop an application by applying appropriate data structures and algorithms (C6,PLO3)
2. The marks allocated for this assignment is 30%.
3. You are required to refer question in Page 2
4. The general format of the written report is as follows:
 - Cover page based on ISO format
 - Font size: 11
 - Font: Arial
 - Justified
5. Save file with the student's name and section no, the name of the assessment and the subject code as the file's name. (*Example of the file's name: AHMAD ALBAB, MUHD AFIF_PROJECT_SECTION1*)
6. You must submit your assignment ONLINE (**PDF file format**) through Learning Management System (LMS) by 6th November 2023 before 5.00pm.

1. Project Objectives

1. To achieve the following Learning Outcome (CO):
CLO 3 – Develop an application by applying appropriate data structures and algorithms.
2. To design Abstract Data Types (ADTs) (with UML diagrams, flow chart, specifications and implementation).
3. To present a complete report (documentation) detailing the whole process of completing the project and final result (output).

2. Important Details

- You are allowed to form a group of 3/4 students.
- This project contributes to 30% of your total coursework.
- The group leader is required to delegate tasks to group members accordingly.

3. Question

Your task is to design and develop a Java application for a Theme Park Ride Ticketing System with three counters, where Counter 1 and Counter 2 are 'Express' counters, and Counter 3 is 'Regular' counters. The system should efficiently manage customer ticket purchases, ticket counters, and generate receipts for customers. The details are as follows:

Phase 1 (LinkedList)

1. Create the following classes named `CustomerInformation` (use appropriate attributes such as `custId`, `custName` and `counterPaid`) and `TicketInformation` (with the instance variables such as `ticketId`, `rideName`, `ticketPrice` and `purchaseDate`). You may add additional attributes to the classes when necessary. Relate these classes using *has-a* relationship in which it keeps the list of items purchased by the customers.
2. Store all customer information in a sequential list named `customerList`. Please be noted that express counters (counter no. 1 and counter no.2) are for customers who purchase max of 5 tickets ONLY.

Phase 2 (Queue, Stack)

3. If a customer buys 5 tickets or less, he/she can go to Counter 1 or 2 to make a payment. The first customer who buys 5 tickets will be directed to Counter 1, the second customer who buys 5 tickets or less will be directed to Counter 2, the third customer who buys 5 tickets or less will be redirected to Counter 1 again and so on. Customers who buy more than 5 tickets, they will be redirected to Counter 3. This process will be repeated until no more records (customer and tickets purchased) are in the input file.
4. Create 3 queue objects and store list of customer information and items purchased to be paid for each counter (Counter 1, 2 or 3) respectively. Then, display the information related to the customer, items purchased and total amount paid (receipt) from each queue accordingly.

5. 5 customers at Counter 1 will make payment first, then the rest 5 customers will replace their place, followed by 5 customers at Counter 2 make payment and so forth. Then repeat again. Store the paid customer in a stack named `completeStack`. Display each customer information and the amount purchase for each customer.

Other requirements:

Your system should be able to read data from file(s). The expected input is:

- Input file to store at least 100 customers who will use Theme Park Ride Ticketing System.

And your system should have the following basic functions:

- Invoke no-args/normal constructors.
- Add new customer line up in a queue/list.
- Remove the respective customer/items from queues when necessary.
- Display customer details and items purchased after he/she makes payment.

4. Innovative and Creativity

This group project will be evaluated based on the logic and functionality of the system as well as the creativity shown in the application. Any related functions are allowed to be added. You can customize your application using the GUI accordingly.

5. Presentation (Week 13, 6-10 November 2023)

Each group must present their Java application based on a given schedule.

6. Report

For this project, you have to submit

- A. A report which contains the following items:

Cover page

Table of contents

1.0 Introduction

2.0 Analysis

3.0 Design of your ADTs using UML Class Diagram and show their relations

Design of the appropriate data structure(s) class (if necessary) or you may import from Java generic class.

4.0 Java code and sample input and output.

5.0 Lessons learned, reflective experience towards completing the individual/overall tasks and Conclusion (Prepared by each group member).

References (APA format)

Example:

1. Joy, B., Steele, G., Gosling, J., & Bracha, G. (2000). *The Java language specification*.

- B. Softcopy upload (.java, .class, files), in github and share the github link in your project documentation.

(TOTAL: 100 MARKS)