

Seneca College

Applied Arts & Technology
SCHOOL OF COMPUTER STUDIES

JAC444

Submission date:12-06-2022

Date:30-05-2022

Workshop 2

Workshop Header (to be included with every file)

/*****

Workshop #

Course:<subject type> - Semester

Last Name:<student last name>

First Name:<student first name>

ID:<student ID>

Section:<section name>

This assignment represents my own work in accordance with Seneca Academic Policy.

Signature

Date:<submission date>

*****/

Code Submission Criteria:

Please note that you should have:

- Appropriate indentation.
- Proper file structure
- Follow java naming convention
- Document all the classes properly
- Do Not have any debug/ useless code and/ or files in the assignment

Deliverables and Important Notes:

All these deliverables are to be uploaded on the blackboard once done.

• You are supposed to create video of your running solution for each task along with demo. It should include voice over explaining the logic and code. You can use diagram like flow charts to aid your explanation. **(40%)**

o Screen Video captured file should state your last name and id, like Ali_123456.mp4 (or whatever the extension of the file is)

• A word/ text file which will reflect on learning of your concepts in this workshop. Also include the instructions on how to run your code. **(30%)**

Seneca College

Applied Arts & Technology
SCHOOL OF COMPUTER STUDIES

JAC444

Submission date:12-06-2022

Date:30-05-2022

o Should state your Full name and Id on the top of the file and save the file with your last name and id, like Ali_123456.txt

- Submission of working code. **(30%)**
- Make sure you follow the “Code Submission Criteria” mentioned above.
- You should zip your whole working project to a file named after your Last Name followed by the first 3 digits of your student ID. For example, **Ali123.zip**. If the zip file is too large to upload on BB , you can upload the video portion on Jac444-NBB team under files/workshop submission/w01
- Your marks will be deducted according to what is missing from the above-mentioned submission details.
- Late submissions would result in additional 10% penalties for each day or part of it.
- Remember that you are encouraged to talk to each other, to the instructor, or to anyone else about any of the assignments, but the final solution may not be copied from any source.

Academic Policies:

Most of the materials posted in this course are protected by copyright. It is a violation of Canada's Copyright Act and [Seneca's Copyright Policy](#) to share, post, and/or upload course material in part or in whole without the permission of the copyright owner. This includes posting materials to third-party file-sharing sites such as assignment-sharing or homework help sites. Course material includes teaching material, assignment questions, tests, and presentations created by faculty, other members of the Seneca community, or other copyright owners.

It is also prohibited to reproduce or post to a third-party commercial website work that is either your own work or the work of someone else, including (but not limited to) assignments, tests, exams, group work projects, etc. This explicit or implied intent to help others may constitute a violation of [Seneca's Academic Integrity Policy](#) and potentially involve such violations as cheating, plagiarism, contract cheating, etc.

These prohibitions remain in effect both during a student's enrollment at the college as well as withdrawal or graduation from Seneca.

Seneca College

Applied Arts & Technology
SCHOOL OF COMPUTER STUDIES

JAC444

Submission date:12-06-2022

Date:30-05-2022

Task

This application is for selling ticket to customer in local cinema house. The Cinema have 2 theaters and hold movies and live dramas. The Cinema has 30 seats in each theater and have 2 seat types (Regular and VIP)

Write a class **Customer** which records basic information such as *customerID*, *Name*, *Address*, *DOB*, *rewardPoints* and *joiningDate*.

Create another class **Ticket** which records *ticketNumber*, *eventTitle*, *eventType*, *theaterNumber*, *price*, *seatRow*, *seatNumber*, *seatType*, *EventDate* (with time) where this will be the date and time the event is, *EventDuration*, *MiscInformation* (specific instructions e.g. Ticket is non-refundable).

Create **TicketSale** class which can have *Customer* object to which ticket was sold, *Ticket* object, *numberTicketSold*, *maxNumberTicket* and *Date*.

All classes should have appropriate constructors, *toString()*, *displayDetails()* and setters and getters methods if needed e.g *rewardPoints* will not need set method. Both *customerID* and *TicketID* should be autogenerated (use variable to keep track of last id). When the customer joins the Cinema, 100 points are added automatically. Additionally, *Customer* class should have *calculateReward()* method to calculate rewards based on following criteria

- Add 100 points if its customer birthday and wish customer "Happy Birthday, You get 100 points gift from us".
- Add 1000 points if customer have anniversary with Cinema. You can use *java.util.Date*, *LocalDate*, *Calendar* classes if needed, to get current date and check if it is year since joining. Wish Customer on the anniversary and thank for remaining loyal customer. Create a separate method to calculate anniversary and birthday and display wishes.
- Add 10 points on each ticket customer buys.

TicketSale class should have *ticketSale()* method to return *TicketSale* object with customer and ticket updated. The sell Ticket will first check for houseful before adding details. The class also needs *IsHouseFull()* method to check and return true if house full and *displayMessage()* method to display msg like "All ticket sold", "Ticket Sale is open", "Ticket Sale closed for Day". The class should also have Calculate methods (to get number of ticket sold, ticket remaining, *eventType*

Seneca College

Applied Arts & Technology
SCHOOL OF COMPUTER STUDIES

JAC444

Submission date:12-06-2022

Date:30-05-2022

and seatType wise ticket distribution) and displayStats() method to display the sale details and menu() (Optionally you can have under one sale multiple tickets).

The main class will be used to bring everything together (to test your classes and functionality). declare some ticketSale, Customer and Ticket variables in main to demonstrate different functionalities using the methods provided in classes. **You are not required to use collection.** Use TicketSale, Customer and ticket type array to have multiple tickets, customer and sale. Display menu to perform different operations.

For demo purpose the program starts by presenting appropriate menu hierarchy to for example

- Add Customer (sub menu to enter details).
- Generate tickets for events (sub menu to enter details).
- Sell Ticket (sub menu to enter customer id and ticket id),
- Display Customer details (submenu to enter customer id),
- Display Sale Stats,
- Search options based on seat type, event type or event title

The menu will repeat for same number of times as you have your array size so that different features can be demonstrated. You can ask in beginning of programming, number of tickets to set ticket array size and additionally have max_customers to set customer size.

Appropriate access specifiers (private, public) and modifiers (final, static) should be used. You may add more methods or classes for computation work if required. Students may use Enum for Event Type and Seat Type (but compulsory). Students may add other details but do put that in reflection. Menu is just as an example; Students are free to change menu items and their hierarchy.

Basic input validations should be considered (separate util class can be used). Output presentation is to be decided by student as well.

Minimum number of classes: 4 (Customer, Ticket, TicketSale, TicketApp(main class))

Optional Classes: Utility, UserInterface (user interaction separate from main)

Optional Additional Classes: for enthusiastic developers (Reward Class: Instead of integer type rewardpoint in customer class , it can be object with rewardpoint and rewardtype (e.g. blue, silver, gold, platinum) as attributes and methods like redeem (ask for number of points to redeem with some threshold to start with and print msg), collect (add points), changeType (change type of loyalty based on reward collected).