1

Notes

Create a single solution/folder to store your source code in a week.

Then, create a project/sub-folder to store your source code for each assignment.

The source code in an assignment should have at least three files:

- A header file (.h): struct definition, function prototypes/definition.
- A source file (.cpp): function implementation.
- Another source file (.cpp): named YourID_Ex01.cpp, main() function. Replace 01 with the id of an assignment.

Make sure your source code is built correctly. Use many test cases to check your code before submitting it to Moodle.

3 Assignments

It is an individual assignment.

Duration: 60 minutes.

Submit to Moodle on time.

Submission file: **<StudentID>_<SelfEvaluatingPoint>.zip**. Ex: 21127001_75.zip

In cinema company X, there are 3 kinds of film tickets.

1. For a Regular Ticket, the following info needs to be stored: film title, start time (hh:mm), room name, price of food and drink, and a ticket factor (default 1.0).
2. For a Combo Ticket, the following info needs to be stored: film title, start time (hh:mm), room name, price of food and drink, and a ticket factor (default 1.5).

Given that:

1. Price of a Regular Ticket = ticket factor * base price (default 80.000) + price of food and drink.
2. Price of a Combo Ticket = ticket factor * base price (default 80.000) + price of food and drink. A 20% discount on food and drink is applied.

Students are asked to apply four characters of OOP, including data hiding, encapsulation, inheritance, and polymorphism, to perform the following tasks:

1. Write methods that allow users to input a list of tickets and store them in a single array. **(2.5 points)**
2. Write methods to sort the list of tickets in ascending order of ticket type (regular ticket first, combo ticket later) and then in descending order of ticket price (if two tickets are the same type). Then write methods to save the sorted list to a text file. **(2.5 points)**
3. Write methods to summarize the list of tickets in 24 time ranges of a day. Display the summary in the console. **(2.5 points)**
4. Draw a class diagram to show classes used in the exam, their attributes, their methods, and their relationship. **(2.5 points)**

Range	**#Regular**	**#Combo**
1	0	2
…	…	…
23	5	6