COMP642 Advanced Programming

Object Oriented Programming Assignment 1

Worth:	15%
Due:	**Friday, 11 August 2023 5:00pm.**
Late Penalty:	Work not received by the due time attracts an immediate penalty of up to 25% of the marks available. No work will be accepted after **Sunday 13 August 2023 5:00pm**.
Submission:	Zip your completed files and submit the .zip through the link on COMP642 LEARN page.

Design and implement a system to manage group exercise classes in a gym. The goal is to create a system that efficiently manages class enrolments, tracks attendance, and calculates payments for gym members participating in group exercise sessions.

Required Classes

The **GroupExercise** Class:

- The GroupExercise class represents a single group exercise session at the gym.
- Attributes:
  - The name of the group exercise class.
  - The trainer assigned to conduct the class (an object of the Trainer class).
  - The maximum capacity of the class.
  - A list of participants (objects of the Member class) who have enrolled in the class.
  - A list of gym members who are on the waitlist for the class.
  - The fee amount for the class.
  - A list of gym members (objects of the Member class) who have checked-in for the class.
- Methods:
  - Enrols a gym member into the group exercise class. If the class is full, the member will be added to the waitlist.
  - Removes a gym member from the enrolled list.
  - Displays all gym members currently enrolled in the group exercise class.
  - Assigns a trainer to conduct the group exercise class.
  - Returns the number of gym members currently enrolled in the class.
  - Returns the number of available slots for enrolment in the class.
  - Sets the fee amount for the class.
  - Calculates and returns the total payment received for the group exercise class based on the number of enrolled members and the class fee.
  - Marks a gym member's attendance for the class.

- Calculates and returns the attendance percentage for the class, representing the ratio of members checked-in to the total number of enrolled members.

The **Member** Class:

- The Member class represents a gym member.
- Attributes:
  - The full name of the gym member.
  - A unique membership number for the gym member.
  - A list of group exercise classes (objects of the GroupExercise class) in which the member is enrolled.
- Methods:
  - Books enrolment in a group exercise class. If the class is already full, the member will be added to the waitlist.
  - Cancels enrolment in a group exercise class.
  - Displays all booked group exercise classes.

The **Trainer** Class:

- The Trainer class represents a gym trainer.
- Attributes:
  - The full name of the trainer.
  - The specialisation or expertise of the trainer.
  - A list of group exercise classes (objects of the GroupExercise class) assigned to the trainer.
- Methods:
  - Displays the list of group exercise classes assigned to the trainer.
  - Adds a group exercise class to the list of classes assigned to the trainer.

The design of the classes must adhere to the object-oriented characteristics of abstraction and encapsulation. All attributes of the class must be private but with public getter and setter methods. Each class should contain:

- Private data members
- Getter and setter methods
- __init__() method
- Methods specific to the class
- __str__() method

Write a driver program to simulate the management of group exercise classes in a gym. You do not need to provide a graphical user interface for this program; just use input and print statements. The driver program must be able to perform the following functionalities:

1. Create 2 GroupExercise objects, 5 Member objects and 2 Trainer objects.
2. Assign a trainer to each group exercise class.
3. Set the class fee for each group exercise class.
4. Set up specific member booking for a group exercise class.

5. Cancelling a specific member’s group exercise class.
6. Record a specific member checking in to a group exercise class.
7. Display the list of enrolled participants for a group exercise class.
8. Display the waiting list for a group exercise class.
9. Display the available slots for a group exercise class.
10. Display the number of participants enrolled in a group exercise class.
11. Display the number of wait list participants in a group exercise class.
12. Display the number of attendees for a group exercise class.
13. Display the attendance percentage for a group exercise class.
14. Display the total payment collected for a group exercise class.
15. Display the list of group exercise classes for which a specific member is enrolled.
16. Display the list of classes offered by a particular trainer.

Marking

Criteria	Marks (out of 100)
GroupExercise Class	**32**
Member Class	**9**
Trainer Class	**7**
Driver	**32**
Coding style, comments, clear logic	**20**
Total	**100**

Marking Criteria

Classes and Driver

Percentage of mark for each criterion is based on the following:

- all requirements are met (81% – 100%)
- some requirements are met (51% - 80%)
- minimum requirements are met (1% - 50%)

Coding Style, Comments and Clear Logic

- good coding style, comments added where relevant and clear logic (16 -20)
- acceptable coding style, some comments added, and logic makes sense (10 – 15)
- acceptable coding style, poor comments, and poor logic (1 – 9)