Collage of Compu
Computer Science Department

Second Semester 1438- 1439

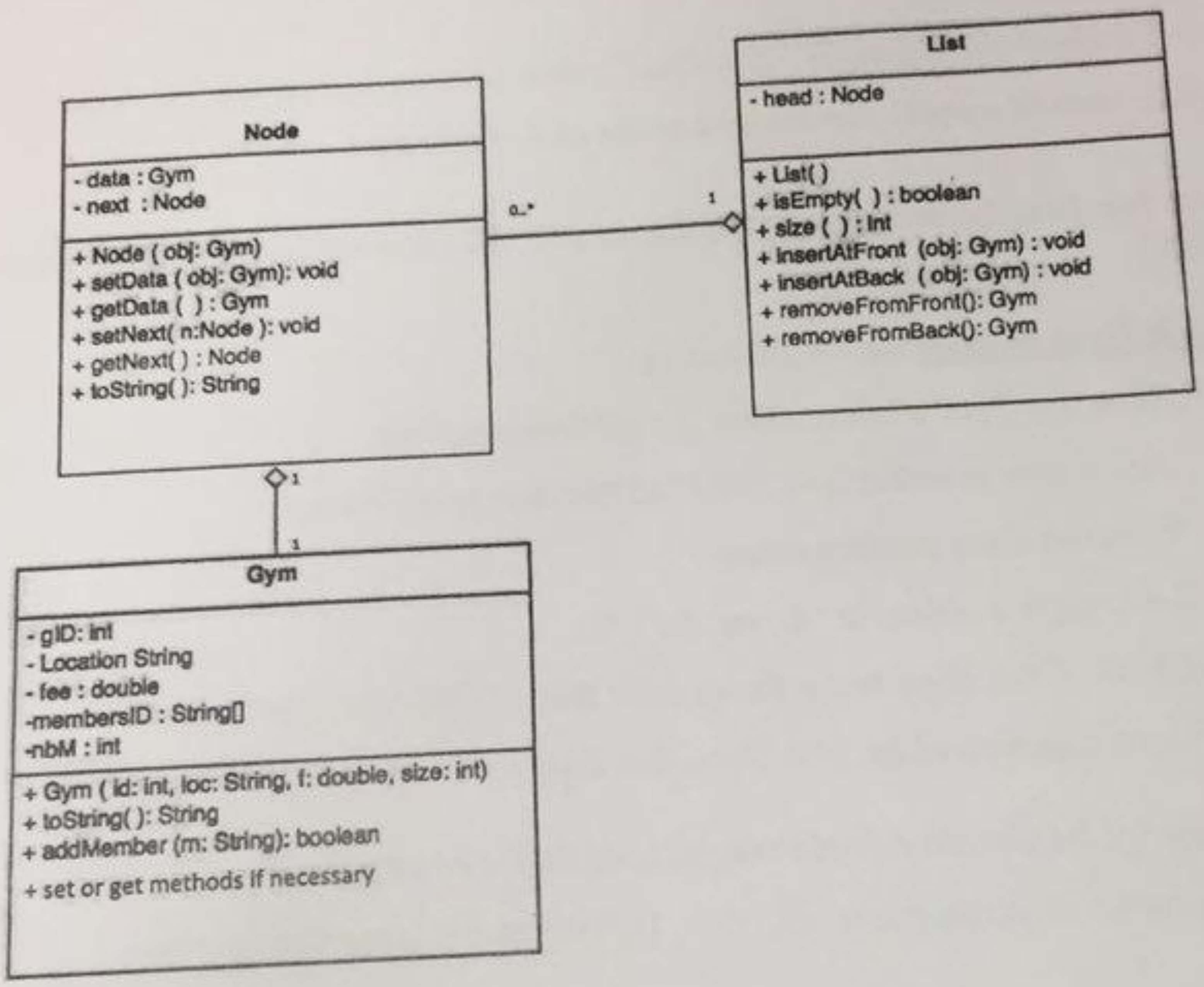
CSC113 Final Lab

Serial#:

ID#:
PC#:

Name:
Lab#:

Important: Create Folder in the desktop with your name (FirstName_LastName) and save all your work on it.



Consider the previous UML Diagram

The classes List and Node have already been implemented for you, **you need to use them properly.** you can find them in the folder CSC113 final lab in the desktop.

1 . Define a new checked exception of type ***InvalidFeeException.***

2 . **Implement Class Gym**

Attributes

- ***gID: int***: The id of the Gym.
- ***location: String***: The location of the gym. •
- ***fee: double:*** gym's fee.
- ***membersID: String[]:*** An array containing id of a gym's members.
- ***nbM: int:*** An integer referring to the first empty entry in the membersID.

Methods

- *Gym(id: int, loc: String, f: double, size: int)*: Constructor to initialize the Gym information.
  Hint: size is the length of the array.
  - An ***InvalidFeeException*** should be generated and caught in the same environment if the fee is less than zero or more than 5000 allowing the user to re-enter the fee.
- *toString()*: **String** Returns the Gym's information (including all membersID) in a string.
- ***addMember(m: String): boolean***: Adds the received member's ID to memberID array. Return true if member id is added successfully, false otherwise.

3. **Create a new Application class with main method to perform the following:**

   a. Using the class List, create a stack of *Gym* named ***gList***.

   Hint: In Stack, the order is Last In First Out (LIFO).

   b. **Add three Gyms** to the list. Ask the user to enter their information. You need to use the appropriate methods.

   c. check if there exists a gym in "*Alnakheel*", then print an appropriate message.

   Hint: gList should remain unchanged at the end of this part

   d. **Create two files for writing:** a text file named *"All Gyms.txt" and* an object file named *"Gyms.dat"*.

   e. For each Gym in *gList* do the following:

   i. Save the Gym's information in "*All Gyms.txt" file.*

   ii. Add a new member's id "*555*" to the *membersID* array.

   iii. Print the *Gym's* information.

   iv. Save *Gym's* object in *"Gyms.dat" file.*

   v. Check if the *Gym* has a fee greater than 2000SAR, insert it in new list name *expensiveList*, else store that *Gym* in an array called ***cheapArray***.

   Hint: gList should remain unchanged at the end of this part.

   f. Then, print the information of all *Gyms* that are in the array cheapArray.

Note:

- **Your main() should catch the possible exceptions that may occur during processing the files.**