

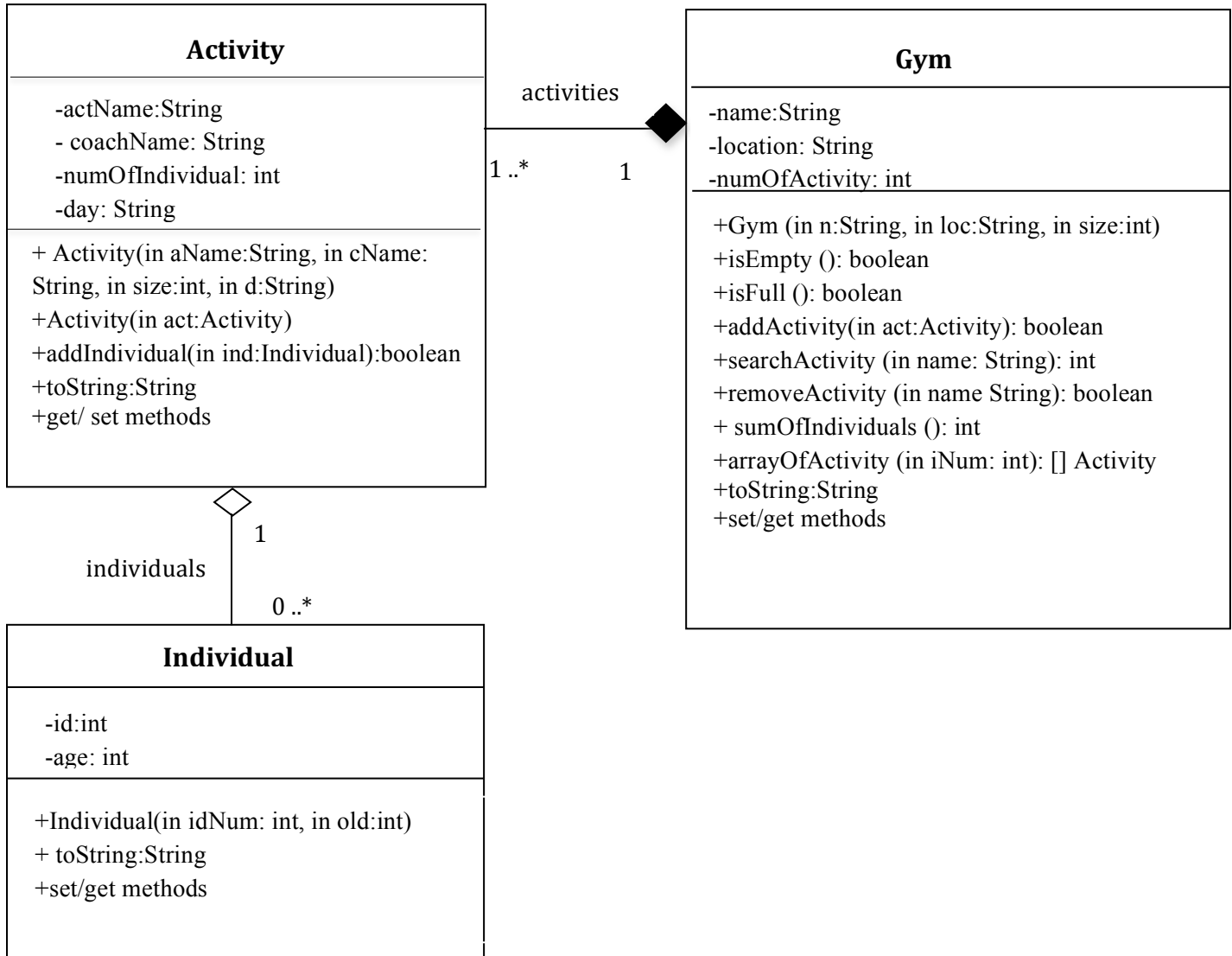
KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
Computer Science Department

CSC 113: Introduction to Programming II

Lab_Sheet#3

2nd Semester 1438

Write the code for the following classes plus its main where:



Activity:

- **actName:** name of the activity.
- **coachName:** name of the coach.
- **numOfIndividual:** number of individuals registered for this activity.
- **day:** the day of the activity.
- **And other attribute(s) deduced from the UML diagram.**
- **addIndividual(in ind:Individual):boolean** : if the array **individuals** is not full, add the given individual to the first empty location in the array **individuals**.
- **toString () :String** : returns a string containing all information of the activity.

Gym:

- **name:** the college name.
- **location:** the location of the gym.
- **numOfActivity:** the number of activities currently given in the gym.
- **And other attribute(s) deduced from the UML diagram.**
- **isEmpty (): boolean** → returns **true** if there are no activities in the gym.
- **isFull (): boolean** → returns **true** if the gym's list of activities are full.
- **addActivity (in act: Activity): boolean** : adds the given activity to the first empty location in the array **activities**.
- **searchActivity (in name: String): int** → returns the index of the activity having the specified name.
- **removeActivity(in name: String): boolean** → deletes the activity with the specified name (replace the deleted one with the last element in the array or shift left).
- **sumofIndividuals (): int** → returns the summation of all the individuals registered in the gym..
- **arrayOfActivity (in iNum:int): [] Activity** → returns an array containing all activities that have a number of individuals > **iNum**.
- **toString () :String** : returns a string containing all information of the gym.

Individual:

- **id**: the individual id.
- **age**: the individual's age
- **toString () :String** : returns a string containing the information of the individual.
- **Main() :**
 - Create an object of gym named **Fitness** that is located in Riyadh and can hold a maximum of 5 activities.
 - Add 2 activities to **Fitness** gym, where each activity can have a maximum of 8 individuals.
 - Add 2 individuals to each activity.
 - Display info of **Fitness** Gym.

Then display the following menu:

1) Add a new activity to the gym:

- a) Ask the user to enter the activity name, coach name, number of individuals and the day of the activity.

Note: If the addition was successful, print a confirmation message that the activity was added successfully, otherwise, print a message that indicates the unsuccessful addition operation.

2) Delete an activity from the gym, knowing its name.

Note: If the deletion was successful, print a confirmation message that the activity was found and the deletion was successful, otherwise, print a message that indicates the unsuccessful deletion operation.

3) Display the number of individuals for a given activity, knowing its name.

Hint: Use the **searchActivity** method

4) Find the sum of all the individuals registered in Fitness gym.**5) Display the names of all activities in Fitness gym which have number of individuals > 6****6) Exit the program.**

The menu should continue to appear until the user wishes to exit the program.