

## Lab Assignment 2

### Question 2,

Membership system for gym & health club

Book different classes such as yoga or spinning

Members can either be juniors or adults

Adults can't book junior classes

Juniors can't book adult classes

Memberships can be either Gold, Silver or Bronze tier

Our first class would be the for all the classes the gym/health club offers. Let's call it 'Classes'. This class will hold all the classes offered in an array, basically like a list of everything that can be booked. This class will also make sure to tag each of the classes as for either adults or juniors. This will help us make sure that classes aren't booked interchangeably between adults and juniors later on. It will also store tags for which tiers each class is available to take. Such as yoga might be available to all three tiers but spinning is will only have the tags silver and gold to indicate that it is not available to bronze tiers to take. One possible way to store all these classes with their associated attributes would be to make each of these classes an object and store their tags as attributes of the object. We can then access these attributes when needed for comparison. We can make use of an array that stores objects, each member represented as an object.

The second class called 'Members' will help us keep track of all the members in the club. It will keep a list of all the members currently in the club and it will also make sure to tag each member as either a junior or an adult. Making this distinction clear here will help us later when the members try to book classes to attend. It may also be possible to make relations between members, such as, a junior member is likely to be the child of an adult member. Making this relation can help us when we apply tiers to members, as everyone in one family is likely to have the same tier membership. We can store the data on members in the same fashion we did for classes. Inheritance might be a solution to building the relations between members to tie them together.

The third class will be called 'Tiers'. This class will apply tiers to members. The three applicable tiers will be Gold, Silver and bronze based on the

## Lab Assignment 2

membership each member possesses. We need not apply this to every member one by one if we use the previously built relations in the Members class to automatically extend one tier of membership to everyone that has relation to one member. For example, children are unlikely to hold an individual membership and are most likely on their parents plan, in this way children will hold the same tier of membership as their parents, allowing us to use just one parent to extend a particular tier of membership to that persons spouse and children. We can add membership tier as an attribute to the objects created for members.

The last class will be called 'Bookings'. This class will use all the data from the previous three classes to allow members to book classes. When a member tries to book a class, this class will pull the names of all the available classes from the 'Classes' class. Once the user picks a class to book, this (java)class will firstly check whether the member making the booking is an adult or a junior and then the same for the class trying to be booked. If both these tags do not match, the user will be notified that this class is not available to juniors/adults. If the tags do match, we will then check what tier membership the member holds and what tiers is the class being booked available to. If it is available to the tier the member belongs to, the class will be able to be booked. If the member does not have the right tier, he will be notified that this class is not available to him. Another possible addition would be to print each classes associated tags with the class name so that members can see this information before hand when picking a class to book. All the information we need to make the comparisons for this class can be accessed as attributes of the objects stored in the arrays in the previous classes.