Exercises

# Aggregation

Aggregation is a persistent relationship between objects. It has more permanence than an association. In this exercise we will build a relationship between a room and a door. We can say that a Room ***must have*** a Door.

## Exercise 1 Add a Door to your design

Create the following class:



* Set the initial state of doorLocked to true and doorOpen to false.
* CloseDoor() sets doorOpen to false
* LockDoor():bool will set doorLocked to true only if doorOpen is false. It will return doorLocked.
* OpenDoor() will set doorOpen to true only if doorLocked is false. It will return the value of doorOpen
* UnlockDoor() will set doorLocked to false

## Exercise 2 – add the door to the room



* Add a Door variable to your Room
* The constructor in Room must make a Door and store it in door
  + Add the line *door = new Door();* to the constructor
* CloseDoor() should call the CloseDoor() of Room and LockDoor():bool methods in Door and return the value of LockDoor():bool.
* Add a method UnlockDoor(accessLevel:int):bool to Room, which unlocks the door if the users accesslevel meets or exceeds the rooms accesslevel and returns true else it returns false.
* OpenDoor(..):bool should call UnlockDoor(..) in room. It should then call and return the value of the Door’s OpenDoor()
* Create a new method in User called UnlockRoom(Room room). This method should call UnlockDoor(..) in Room

Test your design. Can the lecturer unlock the door for the student?