Subclasses and Interfaces

1) Write a class called Room, which has three private instance variables:
- a double width, representing the width of the room in feet,
- a double length, representing the length of the room in feet, and
- an int floor, representing the building floor that the room is on.

a) Write a default constructor for the class Room that sets the width to 10, the length to 12.5, and the floor to 1.

b) Write get and set methods ("getters" and "setters") for the three instance variables. For the set methods for the width and length, only positive values should be set. If the input is 0 or a negative number, the variables should not be changed.

c) Write a constructor for the class Room that takes in two double parameters and an int, and sets length to the larger double, width to the smaller double, and floor to the int. Use the setters from part (b).

d) Override the default toString method for the class Room, so that it a String with the form "length x width on floor floor". For example, calling toString on the instance of Room created by the default constructor from part (a) would return the String "12.5 x 10, floor 1".

e) Implement the Comparable interface for the class Room. The method CompareTo should be written so that an array of Rooms will be sorted first by floor, then by length, and finally by width. Test this method by creating an arrays of Rooms, and sorting them using Arrays.sort(...).

f) Write a subclass of the class Room called Classroom, which also has a private instance variable of type int called numStudents, representing the maximum number of students that the classroom can hold.

g) Write a constructor for Classroom which takes in two double variables and two ints. The instance variable length should be set to the larger double, and the width should be set to the smaller double as in Room. The instance variable floor should be set to the first int, and the instance variable numStudents should be set to the second int. Leave the instance variables as private in Room and use setters to access them.

h) Write a toString method for Classroom which uses the toString method for Room, followed by an additional String ", capacity = numStudents students", where numStudents is replaced by the instance variable value.

i) Write a static method which takes in an array of Rooms and returns an array of Classrooms that contains exactly those Rooms in the input array that are also Classrooms.