



Computer Science Department
Comp1331 Project

Individual work project.

Due Date: Sun. 02/6/2024 by 11:59 pm on **ITC**

In this project, you are required to write a project for a hotel management system: Phase 2 will be as follows:

Write a java program that keeps displaying the main menu containing the following options:

- A. Show All Rooms
- B. Show Available rooms (isOccupied = false)
- C. Change room details
- D. Search by room type
- E. Search by capacity
- F. **Sort and display rooms by pricePerNight ascending using your own sort method.**
- G. **Reserve a specific Room, set isOccupied=true, and input the customer first and last name in an array.**
- H. **Read room details from a file, update rooms based on the file. The format of each line in the file should be as follows: roomNo RoomType isOccupied Capacity pricePerNight. For example:**
201 Standard false 9 250
- I. **Save rooms data to file (No need to store floor number as it can be extracted from room number). The format of each line in the file should be similar to the format of the reading file.**
- J. Quit

1) Class name: Room.java.

a) A room has the following details

- (1) Integer Room number
- (2) Integer Floor number (from 1 to 5)
- (3) String Room Type (Standard, Family, Suite)
- (4) Boolean isOccupied
- (5) Integer capacity
- (6) Double pricePerNight
- (7) Rooms on the first 20 rooms (1 to 20) are standard, rooms: 21 to 40 are Family, and rooms: 41 to 50 are Suite. All rooms have isOccupied = false, and capacity = 2. **The price per Night should be a random integer between 0 and 200 (stored as a double).** Room numbers are numbered from 101 to 110 with floor number=1, 201 to 210 with floor number =2, 301 to 310 with floor number =3, 401 to 410 with floor number =4, and 501 to 510 with floor number=5.
- (8) **Array of Customer name staying in the room (array size is 2: the first element is the first name and the second element is the last name).**
- (9) **Customer name is present only if isOccupied =true for a specific room.**

b) Write appropriate setters and getters for the room class, and toString method.

c) Write appropriate methods to validate correct room numbers, room types, and capacities (assume valid capacity is from 1 to 10).

2) Class name: RoomManagement.java: Room Management System which includes a one dimensional array of rooms of the Room Class (array size = 50). The Room management system allows the behavior of the menu described above. The Room management system allows the following behavior:

- a) Show All Rooms
- b) Show Available rooms (isOccupied = false)

- c) Change room details: to allow to change the following room details at the same time: room type, isOccupied, capacity, pricePerNight. The user must first enter the room number; after that, the room details are entered from the keyboard.
 - d) Search by room type
 - e) Search by capacity
 - f) **Sort and display rooms by pricePerNight ascending. Do not use any java ready methods for sorting. You must write your own sort method.**
 - g) **Reserve a specific Room, set isOccupied=true, and input from keyboard the customer name to stay in the room. For each customer name, use a String array of size 2. The first element of the array is the first name of the customer, while the second element of the array is the last name of the customer.**
 - h) **Read room details from a file, update rooms based on the file. The format of each line in the file should be as follows: roomNo RoomType isOccupied Capacity pricePerNight. For example:**
201 Standard false 9 250
 - i) **Save rooms data to file (No need to store floor number as it can be extracted from room number). The format of each line in the file should be similar to the format of the reading file.**
- 3) Create a Driver class that allow shows the menu above and creates a RoomManagement object and when selecting Room Management System allows the above behavior. All keyboard input should occur in the driver class.

IMPORTANT: What you need to turn in:

- 1- Your project folder (containing all your project files (*.java files) should be compressed (.rar) and saved as **proj_youridnumber_yourLabsectionnumber.rar** (for example if your student id number is 1221234 and your lab **section** is section 9 then the project folder should be called **proj_1221234_s9.rar**). Turn in your project by replying to the course coordinator's message on itc under the lab meta section and attaching your project .rar file (**proj_youridnumber_yourLabsection.rar**).
- 2- You must include your full name, student id number, and lab section number in a comment at the beginning of each of your .java code files.

Late Projects will NOT be accepted for any reason.

Project Files that are not compressed correctly or cannot be opened will not be graded and will receive a zero grade.