hotel Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In this question, you will implement two methods for a class Hotel that is part of a hotel reservation system. The Hotel class uses the Reservation class shown below. A Reservation is for the person and room number specified when the Reservation is constructed.

public class Reservation

{

private String guestName;

private int roomNumber;

public Reservation(String guestName, int roomNumber) {

this.guestName = guestName;

this.roomNumber = roomNumber;

}

public int getRoomNumber() { return roomNumber; }

public String toString() {

return "[" + guestName + ", " + roomNumber + "]";

}

}

An incomplete declaration for the Hotel class is shown below. Each hotel in the hotel reservation system has rooms numbered 0, 1, 2, . . . , up to the last room number in the hotel. For example, a hotel with 10 rooms would have rooms numbered 0, 1, 2, . . . , 9.

import java.util.ArrayList;

public class Hotel

{

// each element corresponds to a room in the hotel;

// if rooms[index] is null, the room is empty;

// otherwise, it contains a reference to the Reservation for that room,

// such that

// rooms[index].getRoomNumber() returns index

private Reservation[] rooms;

// contains names of guests who have not yet been

// assigned a room because all rooms are full

private ArrayList<String> waitList;

// Constructor - creates a Hotel with 'n' number of rooms

public Hotel(int nRooms) {

rooms = new Reservation[nRooms];

waitList = new ArrayList<String>();

}

// if there are any empty rooms (rooms with no reservation),

// then create a reservation for an empty room for the

// specified guest and return the new Reservation;

// otherwise, add the guest to the end of waitList

// and return null

public Reservation requestRoom(String guestName)

{

/\*\*\* <<< to be implemented in part (a) >>> \*\*\*/

}

// release the room associated with parameter res, effectively

// canceling the reservation;

// if any names are stored in waitList, remove the first name

// and create a Reservation for this person in the room

// reserved by res; return that new Reservation;

// if waitList is empty, mark the room specified by res as empty

// and return null

// precondition: res is a valid Reservation for some room

// in this hotel

public Reservation cancelAndReassign(Reservation res)

{

/\*\*\* <<< to be implemented in part (b) >>> \*\*\*/

}

public String toString() {

String s = "";

for (int i = 0; i < rooms.length; i++)

s += rooms[i] + " ";

s += "Waitlist: " + waitList;

return s;

}

}

/\*\*\* <<<< THIS IS A COMPLETE TESTER CLASS BELOW >>>

\* Output from Tester:

\*

\* Reserve Amanda [Amanda, 0]

\* Reserve Ben [Ben, 1]

\* Reserve Cate [Cate, 2]

\* Reserve Don null

\* Reserve Euginia null

\*

\* Hotel: [Amanda, 0] [Ben, 1] [Cate, 2] Waitlist: [Don, Euginia]

\*

\* Reassign Rm 2 [Don, 2]

\* Reassign Rm 0 [Euginia, 0]

\* Reassign Rm 1 null

\*

\* Hotel: [Euginia, 0] null [Don, 2] Waitlist: []

\*

\* Reserve Frank [Frank, 1]

\* Reserve Gabrielle null

\*

\* Hotel: [Euginia, 0] [Frank, 1] [Don, 2] Waitlist: [Gabrielle]

\*\*\*/

public class TestHotel

{

public static void main(String[] args)

{

Hotel hotel = new Hotel(3); // Hotel with 3 rooms (Small Hotel!!!)

System.out.println("Reserve Amanda " + hotel.requestRoom("Amanda"));

System.out.println("Reserve Ben " + hotel.requestRoom("Ben"));

System.out.println("Reserve Cate " + hotel.requestRoom("Cate"));

System.out.println("Reserve Don " + hotel.requestRoom("Don"));

System.out.println("Reserve Euginia " + hotel.requestRoom("Euginia"));

System.out.println();

System.out.println("Hotel: " + hotel);

System.out.println();

System.out.println("Reassign Rm 2 " + hotel.cancelAndReassign(new Reservation("Cate", 2)));

System.out.println("Reassign Rm 0 " + hotel.cancelAndReassign(new Reservation("Amanda", 0)));

System.out.println("Reassign Rm 1 " + hotel.cancelAndReassign(new Reservation("Ben", 1)));

System.out.println();

System.out.println("Hotel: " + hotel);

System.out.println();

System.out.println("Reserve Frank " + hotel.requestRoom("Frank"));

System.out.println("Reserve Gabrielle " + hotel.requestRoom("Gabrielle"));

System.out.println();

System.out.println("Hotel: " + hotel);

System.out.println();

}

}