

<b>King Saud University</b> <b>College of Computer and Information Sciences</b> <b>Computer Science Department</b>	
<b>CSC 111</b> <b>Introduction to Programming with Java</b>	<b>First Semester</b> <b>1440-1441</b>

**Project**  
**Hotel Reservation System**  
**Due Date 10/12/2019**

---

## Problem Description:

We want to implement a *hotel reservation system* that is used by **guests** and **hotel clerks**. The system should allow Guest and hotel clerk to log in , and based on the log in , it display a specific menu of operations that can be done by either guest or hotel clerk.

These operations include: making reservation, Canceling reservation, displaying reservation information and so on.

A **guest** and **clerk** can log on into the system. Each type of user has it's own menu options. If **guest** is logged in then the following menu options is displayed:

- Make a reservation.
- Cancel current reservation
- Display current reservation information

However, if **clerk** logged in then the following menu options is displayed:

- Display a reservation.
- Display all Confirmed reservations.
- Modify/Cancel reservation.
- Count the number of reservations
- Display all due reservations.

Each **Reservation** has the following information:

- Room Type (Single bedroom – Twin Bedroom – Family Bedroom)
- Check-In Date.
- Check-Out Date.
- Price.
- Reservation Number.
- Reservation states ( Confirmed – Unconfirmed - Canceled).

<b>King Saud University</b> <b>College of Computer and Information Sciences</b> <b>Computer Science Department</b>	
<b>CSC 111</b> <b>Introduction to Programming with Java</b>	<b>First Semester</b> <b>1440-1441</b>

### ***Assumptions:***

- Reservation number is generated only when reservation is Confirmed, this is done when guest enters a valid credit card number (credit card number is a 10 digit number, a special method is used to validate this number ; validation method is explained in **Reservation class**)
- Guest will first select room type, check in date and check out date (for simplicity, guest will first select month then select the day, check out date must be valid)
- Price of the room will be calculated based on a table explained in **Reservation class**.
- There is a counter for the number of reservation that will be incremented whenever a reservation is successfully made and added (regardless of the reservation states).
- The Hotel Reservation system can have up to 100 reservations.

### **Implementation Details:**

In order to implement the Hotel Reservation System you should have two classes: ***Reservation*** and ***Hotel***.

#### **Reservation Class**

This class represents a **reservation**. It holds a reservation's information.

- **Check-in Date:** guest enter month then day of check in , the check in date has the following format DD- MM
- **Check-out Date:** for simplicity, guest will only select the day of the check out , it should not precede the check in day , format DD- MM
- **Room Type:**

Type	code	Price per night
Single	S	150 SR
Twin	T	220 SR
Family	F	400 SR

- **Reservation number:** the guest will confirm the reservation by entering a correct credit card number, method **verifyCard()** is called to validate the entered credit card. If the card is validated then reservation status is set to “ Confirmed” , price of room is calculated and Reservation number is generated.

<b>King Saud University</b> <b>College of Computer and Information Sciences</b> <b>Computer Science Department</b>										
<b>CSC 111</b> <b>Introduction to Programming with Java</b>						<b>First Semester</b> <b>1440-1441</b>				

Otherwise, reservation status is set to “Unconfirmed” , Reservation number is set to "#####" and room price is 0 ,

- **Reservation status:** a reservation is either :
  - “Confirmed” if all information is correct and credit card is valid.
  - “Unconfirmed” if all information is correct **BUT** credit card is invalid.
  - “Canceled “if guest/clerk select to cancel.
- **Price :** total Price of the room for the duration (room cost X total number of nights)

*Reservation Class* contain the following methods:

- **Reservation (rType , indate , outdate , card ) :** A constructor that initializes reservation with the initial values from the user.
- **verifyCard (int card):boolean**
  - a method that is called from the constructor to check if the credit card number is valid or not , if the credit card is valid method will set status to “Confirmed” and return true , otherwise it set status to “Unconfirmed” and return false .
  - This method uses the Luhn algorithm for verify the card number.

### The Luhn Algorithm:

It is a simple checksum formula used to validate identification numbers. Because the algorithm is in the public domain, it can be used by anyone to distinguish valid numbers from mistyped or otherwise incorrect numbers.

### How does Luhn algorithm works?

Consider the example of credit card number “4773100241”.

**Step 1** – Starting from the rightmost digit double the value of every second digit,

Credit card number	4	7	7	3	1	0	0	2	4	1
		X2		X2		X2		X2		X2
		14		9		0		4		2

**Step 2** – If doubling a number results in a two digits number, then add the digits of the product (e.g., 12: 1 + 2 = 3) to get a single digit number.

Credit card number	4	7	7	3	1	0	0	2	4	1
		X2		X2		X2		X2		X2
		14		9		0		4		2
		5								

<b>King Saud University</b> <b>College of Computer and Information Sciences</b> <b>Computer Science Department</b>										
<b>CSC 111</b> <b>Introduction to Programming with Java</b>						<b>First Semester</b> <b>1440-1441</b>				

**Step 3** – Now calculate the sum of all the digits.

Credit card number	4	7	7	3	1	0	0	2	4	1
		X2		X2		X2		X2		X2
		14		9		0		4		2
		5								
Sum	4	5	7	9	1	0	0	4	4	2

$$4+5+7+9+1+0+0+4+4+2= 36$$

**Step 4** – If the total modulus 10 is equal to 0 then the number is valid according to the Luhn formula; else it is not valid.

*Since the sum is 36 which is not a multiple of 10, therefore the credit card number is invalid.*

- **CalculatePrice(): double**
  - A method that is called from the constructor only if the credit card is valid (otherwise the cost is 0). This method calculate and returns the price of the room ( room price X duration ).
- **GenerateReservationNumber():String**
  - A method that is called from the constructor (Only if the credit card is valid), Reservation number is automatically generated as follow :  
[Number of reservation - Random lower case letter – random two digit number – random upper case letter].
- **CancelResrvation():boolean**
  - This method will change the status of reservation to “Canceled” and return true if it is done.
  - Note: if the room is already cancel , display an appropriate message and return false
- **Print():void**
  - This method will display the reservation information in the following format:

```

Reservation number : XXXXX      Reservation Statues :XXXXX
Room type : XXXX              Check in Date : XXXXX
Price :XX.XX

```

- Any necessary setters and getters.

King Saud University College of Computer and Information Sciences Computer Science Department	
CSC 111 Introduction to Programming with Java	First Semester 1440-1441

## Hotel Class

This class represents the Hotel. Class **Hotel** contains an array of objects that holds all the **Reservations** information.

The **attributes** of Hotel:

- **Reservation[] hotel\_Reservation** : an array of objects of type **Reservation** which holds the list of reservations in the hotel.
- **MAX\_SIZE**: a public static final attribute that stores the maximum number of reservations that the hotel can handle

*Class Hotel* contain the following public methods:

- **main():void**
  - The main method will allow different users to log in (guest/clerk) , it should keep asking the user to log in until the user select to exit.
- **login(String typeOfUser) : void**
  - a method that will show different menus based on the user (guest/clerk) .

Guest Menu:	Clerk Menu
<ul style="list-style-type: none"> <li>• Add reservation</li> </ul> <p>then sub menu :</p> <ul style="list-style-type: none"> <li>• Cancel reservation</li> <li>• Display reservation</li> </ul> <p><i>Note: submenu appear only if reservation was added successfully</i></p>	<ul style="list-style-type: none"> <li>• Find reservation</li> <li>• Display all Confirmed reservations</li> <li>• Modify reservation.</li> <li>• Cancel reservation.</li> <li>• Count the number of reservation.</li> <li>• Display all due reservations.</li> </ul>

- The method will keep showing menu until user select to exit.
- **AddReservation():boolean**
  - This method is called by the guest, it will add a reservation by reading the following information: the room type, check in date, check out date, credit card number.
  - The new reservation is added to the end of the array of reservations . This method *returns true* if the add operation was completed successfully, and *false* otherwise. The reservation is successfully added if the array is not full.

<b>King Saud University</b> <b>College of Computer and Information Sciences</b> <b>Computer Science Department</b>	
<b>CSC 111</b> <b>Introduction to Programming with Java</b>	<b>First Semester</b> <b>1440-1441</b>

- **CancelMyResrvation():boolean**
  - This method is called by the guest to cancel the current reservation (it change the status to “Canceled”) and return true , if the reservation is already canceled return false.
- **DisplayMyReservation():void**
  - This method is called by the guest to display the current reservation information.
- **Find(String resNumber):int**
  - The method is called by the clerk to find a specific reservation given its reservation number, the method should return the index of this reservation in the array of reservations , otherwise return -1 if reservation doesn't exist.
- **Display():void**
  - The method is called by the clerk to display the information of all “Confirmed” reservations.
- **Modify(String resNumber):Boolean**
  - The method is called by the clerk, given the reservation number, **the** clerk will update a reservation (room type) .
  - If the reservation is found and updated, return true. Otherwise, return false.
  - *Note: some features of the reservation may need to updated accordingly*
  - *Note: Modify method work on confirm reservation only.*
- **Cancel(String resNumber):Boolean**
  - Given the reservation number, the method will change the reservation status from” Confirmed” to “Canceled” .
  - If the reservation is found and canceled return **true**. Otherwise, return **false**.
- **CountReservations (String status):int**
  - Given the status of reservation ( ” Confirmed” –"Unconfirmed"- “Canceled”), the method will return the total number of the selected reservation status.
- **DisplayDueReservations ():void**
  - Based on the system current date, this method will display all due reservations. (Due reservations are reservation where tomorrow is the **check in** date).

*Note: you can use the code provided at the end to get the system date.*

### **Requirements:**

- You must define the classes **Reservation** and **Hotel**.
- Same as all programming assignments:
  - Use meaningful variable name and good indentation.

<b>King Saud University</b> <b>College of Computer and Information Sciences</b> <b>Computer Science Department</b>	
<b>CSC 111</b> <b>Introduction to Programming with Java</b>	<b>First Semester</b> <b>1440-1441</b>

- You must avoid code duplication, by calling appropriate methods (rather than cutting and pasting code).
- Don't change the names of any class, attribute or method.
- Submit your project even if you are not able to complete all the functions.

Note: If you couldn't implement a method then provide a shell (empty body, just a header) for the method. This will allow us to compile your program and test the components you have implemented.

// This code will Get Current Date and Time as string

```
import java.text.SimpleDateFormat;
import java.util.Date;
public class MyClass{
    public static void main(String args[]) {
        String date = new SimpleDateFormat("dd-MM-yyyy").format(new Date());
        System.out.println("Date is " + date);
    }
}
```

<b>King Saud University</b> <b>College of Computer and Information Sciences</b> <b>Computer Science Department</b>	
<b>CSC 111</b> <b>Introduction to Programming with Java</b>	<b>First Semester</b> <b>1440-1441</b>

**The project has two phases:**

**Phase 1 (Design Phase)**

(Upload UML on LMS before **Thursday 21/11/2019 -11:59 p.m.**)

Design the UML for the classes (**Reservation and Hotel**). Please make it nice and neat.

Note: you can use [www.gliffy.com](http://www.gliffy.com) to draw the UML

On Sunday 24/11/2019 the correct UML will be available on LMS and you will use it to implement your classes (you will implement the correct one).

You will need to finish mapping the UML to the correct code (class, attributes and method headers).

**Phase 2 (Implementation Phase)**

- 1) Complete the implementation of all methods bodies.
- 2) Test your code using the following test cases.

Reservation 1	Reservation 2	Reservation 3
Room Type : <b>Twin</b>	Room Type : <b>Single</b>	Room Type : <b>Family</b>
Check in date : <b>2- 5</b>	Check in date : <b>15- 6</b>	Check in date : <b>1- 12</b>
Check out date : <b>10 - 5</b>	Check out date : <b>27 - 6</b>	Check out date : <b>4 - 12</b>
Card number : <b>7992739875</b>	Card number : <b>5866321747</b>	Card number : <b>4884596271</b>

**Submission:**(Upload your project on LMS before **Tuesday 10/12/2019 -11:59 p.m.**)

- Your project should be submitted on LMS (with your name, section number, serial number and date) that includes your code and report with results ( use the report template )  
*Note: since you are a group, the group leader will upload your work.*
- Discussion will be on Wednesday 11/12/2019
- On the discussion day you need to bring your laptop with the code ready for running.
- You need to be ready 10 minutes early before your time slot (time and date will be announced)
- Please be ready to explain your code and run your code using the test data that will be provided by the TA during your discussion time.