

Computer Science Department
CCCS 220, 2nd Term 2019
Assigned: Thursday, February 07th, 2019
Due: **Thursday, March 7th, 2019 (30-06-1440 H)**
Phase 1

Purpose:

The purpose of this project is to:

1. Introduce Object Oriented Style of programming.
2. Practice your knowledge of design classes in Java.
3. Practice the implementation of various kinds of classes in Java.
4. Organize the data and the functionality into required classes using object oriented concepts.
5. Learn to use and implement String, StringBuilder, File I/O (Reading/Writing from/to files).

Read Carefully:

- This Project is worth 10% of your final grade.
- The project consists of two phases (Phase 1 is worth 4% and Phase 2 is worht 6%)
- **WARNING:** This is an individual project; you must solve it by yourself. Any form of cheating will result in receiving **-4%** (less than zero) in the program. Plagiarism-detection tools will be used for all submissions.
 - The deadline for Phase 1 is by **11:00 PM on Thursday, March 7th, 2019 (30-06-1440 H)**
 - **Note:** once the clock becomes 10:59PM, the submission will be closed! Therefore, in reality, you must submit by 10:58 and 59 seconds.
- **LATE SUBMISSION:** you are allowed to make a late submission, but there is a penalty. If you submit **within 24 hours** of the due date (so on Friday by 10:59PM), you will receive a 25% deduction. You will NOT be able to submit after this date/time.

Deliverables and Blackboard Submission:

- This project must be submitted online via blackboard
- If your file is empty or you upload the wrong file, it will be solely your responsibility, and your grade will be zero.
- **Phase 1 (UML part)**
you need to submit a **UML class diagram** (in pdf, gif or jpeg format) with a brief discussion of such design and should be named as:
CCCS220Project_StudentId_SectionName.pdf
Example: CCCS220Project_1110348_LA.pdf
- **Phase2 (the coding part)**
You must include only the source files in your submission (do not submit any *.class files!) (Your submission must be one zip file).
 - The main file should be on the format CCCS220Project_StudentId_SectionName, if it's not in this format you will lose points.

Project Description:

Task: Hotel Booking System

The software product to be produced is a Hotel Booking System, which will automate the major hotel operations. The first subsystem is a Booking System to keep track of reservations and room availability. The second subsystem is the Tracking and Selling Services System that charges the current room. The third subsystem is a General Invoice Management System, which generates invoices for the hotel' visitors.

System reads all data from a given input files [InputRooms.txt, InputProcedures.txt] and generate result and reports in several output file like [BookingStatus.txt, Sa11167_Guest_Invoice.txt].

More Details are as follows:

The Initial Settings of the Hotel [Rooms and Services]:

Your program will use File I/O to read input from a given file name [InputRooms.txt]. Make sure the file exists or display a message that the file does not exist.

The file contains three types of commands: [AddRoom, AddBungalow and AddService]

You will have to implement methods for these commands in the tester class:

AddRoom– Makes a new Standard Room, which is added to the system. [See InputRooms.txt]

Command **AddRoom** is followed by **3 integers** [the floor number, room number and Price], and **2 strings** [View City or Ocean and Bed type single, double or King] [example from the input file]:

AddRoom	10	101	150	City	single
	Floor	room	Price	View	Bed
	number	number			type

AddBungalow– Makes a new Bungalow Room, which is added to the system. [See InputRooms.txt]

Command **AddBungalow** is followed by **4 integers** [the floor number, room number, Price, number of rooms], and **1 string** [Outdoor type Pool or Garden] [example from the input file]:

AddBungalow	0	10	1500	2	Pool
	Floor	room	Price	number	Outdoor
	number	number		of rooms	type

AddService— Makes a new Service, which is added to the system. [See **InputRooms.txt**]

Command **AddService** is followed by 1 string [Service type *Spa, Laundry, RoomService_Breakfast, RoomService_Lunch, RoomService_Dinner, BabySitter or Gym*] and 1 integer [Service Price] [example from the input file]:

AddService	Gym	100
	Service	Service
	type	price

Front-desk activities of a hotel [Book rooms, Services, Print invoice]: Your program will use File I/O to read input from a given file name [**InputProcedures.txt**]. Make sure the file exists or display a message that the file does not exist.

The file contains three type of commands: [**Reserve, AddServiceVisitor and Print_Invoice**]

You will have to implement methods for these commands in the tester class:

Reserve is followed by the type of Room [*StandardRoom or Bungalow*], all information about the such room, Check In and Check Out dates, type of Visitor [*G for visitor and V for VIP*] and all information about visitor. In case of VIP visitor, the last value represents the *VIP Id*[See the input file].

Case 1 : Reserve StandardRoom

In case of the requested Standard Room is available, the reservation is accepted, makes a new booking and added it to the system. A room is available if a Standard Room with requested features [*Bed type, View*] is available throughout the specified period [*From Check in to Check out period*]. [See **Reserve StandardRoom in InputProcedures.txt**]

Notes:

- For each new Booking, generates a new Booking reference number as follow:
Visitor First Letter + Visitor Id+ Random value (between 0 and 999). Example : S11167260
- Search for the requested room category through all the rooms in the system.
- It's mandatory to use Class Date methods to check that the room is available throughout the specified period
- If the reservation is accepted, add this Booking to the list of Booking of the reserved room.
- Each Booking status must be saved in the **BookingStatus.txt** file as per given sample **BookingStatus.txt** file.

Case 2 : Reserve Bungalow``

In case of the requested Bungalow is available, the reservation is accepted, makes a new booking and added it to the system. A Bungalow is available if a Bungalow with requested features [*Number of rooms, Outdoor type*] is available throughout the specified period [*From Check in to Check out period*]. [See **Book Bungalow in InputProcedures.txt**]

Notes:

- For each new Booking, generates a new Booking reference number as follow : **Visitor First Letter + Visitor Id+ Random value** (between 0 and 999). Example : S11167260
- Search for the requested Bungalow through all the Bungalows in the system.
- It's mandatory to use Class Date methods to check that the Bungalow is available throughout the specified period
- Each Booking status must be saved in the **BookingStatus.txt** file as per given sample **BookingStatus.txt** file.

AddServiceVisitor is followed by 1 integer [**Visitor ID**] and 1 string [**Service type** *Spa, Laundry, RoomService_Breakfast, RoomService_Lunch, RoomService_Dinner, BabySiiter or Gym*] [**See the input file**]. This command will be used to assigned a service to a visitor. [See **Service in InputProcedures.txt**]

Notes :

- From the list of services, search for the requested service in the system using *Service type* to find price
- From the list of Booking, search for the Booking of the visitor who is requested this service using *Visitor ID*.
- Add this service to the list of services of this Booking

Print_Invoice This command will have no other information on the line. It will be used to print ALL invoices details in a separate file for each Visitor, each invoice in a separate file. Invoice details include Visitor information, Standard Room or Bungalow information, Check in date, Number of nights, Room price for the period, and services provided to the visitor.

Notes :

- Before printing invoices, all invoice files must be stored in the system
- It's mandatory to use Class Date methods to compute number of nights
- If the visitor is a VIP visitor, 25% reduction will be directly deducted from his

- Invoice file name must be given as
Visitor First two Letter + Visitor Id+" _Guest_Invoice" + ".txt"
for example A110665_Guest_Invoice
[see All Invoices files For All Visitors]

Further Details are as follows:

You have to create **SEVEN classes** in this program.

1. **Room** class is super class of StandardRoom and Bungalow classes.
2. **StandardRoom** class to store the standard room details
3. **Bungalow** class to store the Bungalow details
4. **Booking** class to store the Booking details
5. **Service** class to store the service details
6. **Visitor** class to store the visitor details
7. **VIP** class is a sub-class of **Visitor**.

Tester class (HotelSystem) to create objects and invoke appropriate methods for program to execute successfully. And You have to create FOUR ArrayList in this program.

ArrayList of Rooms to store the hotel Standard Rooms and Bungalows
[available in InputRooms.txt].

ArrayList of Services to store the available services in this hotel.
[available in InputRooms.txt]

ArrayList of Booking to store the accepted booking.
[From InputProcedures.txt]

ArrayList of File to store all invoice files.

- In the Room class, you have to create an **ArrayList of Booking** to store all accepted Booking for a specific room.
- In the Booking class, you have to create an **ArrayList of Service** to store all provided Services related to a specific Booking.

Important Notes:

- Use of class & object, ArrayList of Object, passing object to method, dynamic binding and Inheritance is mandatory.
- Use and implement Polymorphism and Object Explicit casting.
- Use the instanceof operator.
- Use of Files, Reading/Writing from/on files and String or StringBuilder methods.
- Your program output must be exactly same as given sample output files.
- Your display should be in a readable form.
- Organize your code in separated methods.
- Repeat the program until command=Quit.
- Document your code with comments.
- Use meaningful variables.
- Use dash lines between each method.

Input and Output Format:

Your program must generate output in a similar format to the sample run provided.

Sample input: See sample input files.

Sample output : See sample output files.

Suggestions:

- Read AND fully understand this document BEFORE starting the program!
- Once the solution is 100% clear to you, then begin making your code.

Hope this helps.

Final suggestion: START EARLY

KINGDOM OF SAUDI ARABIA
Ministry of Education
UNIVERSITY OF JEDDAH
Main Campus- Girl's
Faculty of computing and
Information Technology



المملكة العربية السعودية
وزارة التعليم
جامعة جدة
شطر الطالبات