- Go through the problem statement clearly.
- Time limit is 180 minutes.
- Make sure that project is created in eclipse only.
- Create all your java files in package **com** within src folder of eclipse project.
- Make sure that exact class outline is followed as you did in previous assignments.
- You need to zip the eclipse project folder and upload the same in LMS once completed. The project folder will be available in your workspace folder.
- It is mandatory to upload eclipse project and not java files alone, for your code to be assessed.
- Make sure that there is no compilation error in your code before submission. Even if there is minor error, entire solution could be rejected.
- You may refer previous assignments, course content and internet for any reference.

    - Last method of Gulmohar Grand Hotel class is **optional**.

Gulmohar Grand Hotel is planning to automate their processes to manage the operations better.

Mainly there are below operations for which a Java based solution is required:

- Adding customers to the Hotel
- Adding rooms to the Hotel with different room types
- Allocating types of rooms to different customers

Create project Hotel Management System in eclipse.

All classes should be created inside **com** package.

Create class **Customer** with below outline.

Create constructors, generate getters/setters and override required methods with the exact details given in outline.

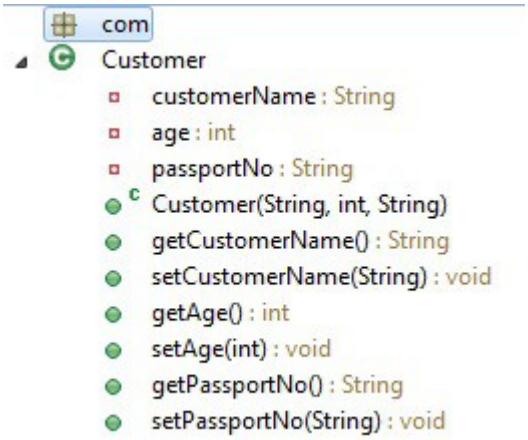
The constructor of Customer class should be of following sequence.

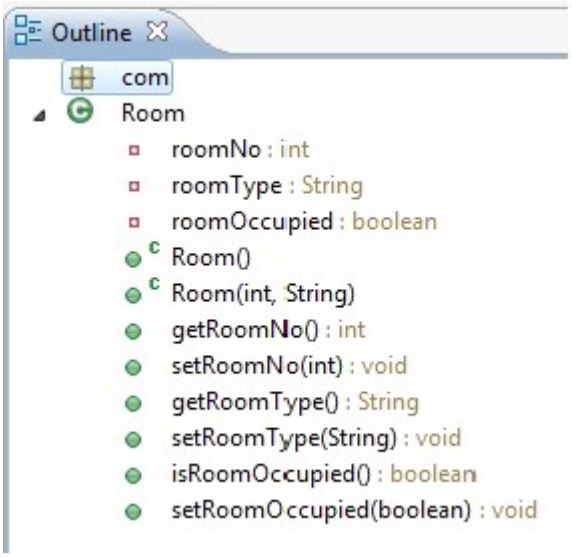
**(String custName,int age,String passportNo)**

Create **Room** class with below outline. Create constructors, generate getters/setters with the exact details given in outline.

Room Type : could be AC or Non AC.

By default roomOccupied should be kept as false.



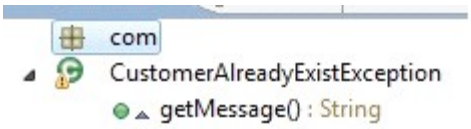

Create a customised checked exception **CustomerAlreadyExistException** class and override getMessage method as given in the outline. The above exception should be thrown whenever same customer is being added once again.

Return below message from getMessage() method:

“Customer Already Exist !! Please recheck before adding new Customer”

**Please copy paste above message properly/don't make a mistake even of a full stop/otherwise**

**code will not compile in iAscert Tool**

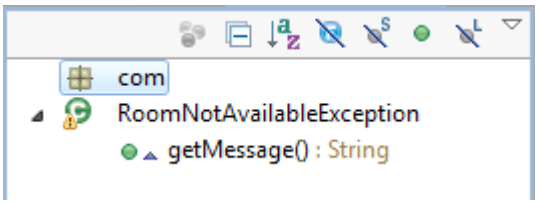



Create a customised checked exception RoomNotAvailableException class and override getMessage method as given in the outline.

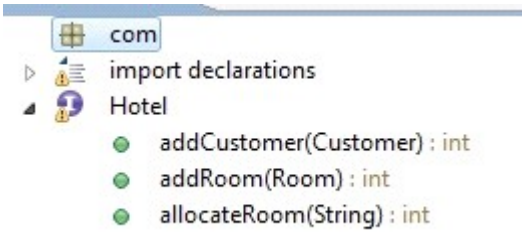
Return below message from getMessage() method:

“Currently Room is not available.It is already occupied by other customer”

**Please copy paste above message properly/don't make a mistake even of a full stop/otherwise code will not compile in iAscert Tool**




Create an interface **Hotel** –
This is a contract to follow the design for building Hotel Management System. Refer below outline



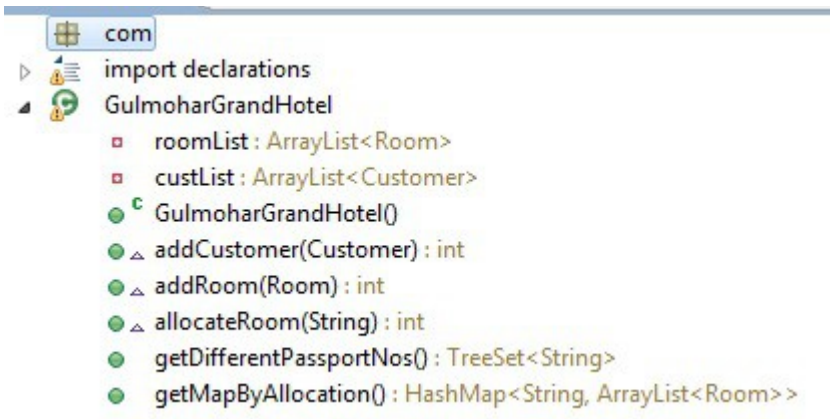

Note In the above interface:

- addCustomer method  throws  CustomerAlreadyExistException
- allocateRoom method throws RoomNotAvailableException

Create a class **GulmoharGrandHotel** with below details.

- Implement Hotel interface and all its methods with appropriate logic.
- GulmoharGrandHotel class should contain ArrayList of Customers and Rooms.
- Should have main method to test the instance methods.

Refer below outline for this class.




Consider below methods in **GulmoharGrandHotel** class:

**addCustomer** – This method will take Customer reference as a parameter. The method should add the Customer to the list. Before the customer is added to the list, the method should check if customer already exists in the system. Combination of “name” and “passportNo” of customers will be used to check if the same customer exists in the Hotel List already. A Customer already exists with the same name and passportNo then, method must throw CustomerAlreadyExistException - message - "Customer Already Exist !! Please recheck before adding new Customer"
. Do not handle the exception there. Please throw the same. If the customer is added successfully then total count of customers should be returned.

**addRoom** – This method will take Room object reference. It will add this object to the list and returns the total number of rooms in the list. This method should not add duplicate rooms in the list. Two rooms with the same roomNo are considered duplicate. The method should return the total number of rooms available in the list even if new room is not added.

**allocateRoom** –This method will take roomType as parameter. The method should search if there exists a room in the list whose roomOccupied flag is false. If so change the flag value to true and return the room number whose flag was changed. If a room could not be found with the provided roomType with roomOccupied flag as false the method must throw RoomNotAvailableException - message - "Currently Room is not available. It is already occupied by other customer"

**getDifferentPassportNos** – This method will return a **TreeSet** of String types, where it will contain unique set of passport numbers(sorted) that Customers of GulmoharGrandHotel have.

**getMapByAllocation(Optional)** – *This method is optional. Attempt this one if you finished implementing above methods.* This method will return **HashMap** with String as key and **ArrayList of Room objects** as value. This **HashMap** should only contain 2 entries:

**Key** - “allocated” **Value** – ArrayList of Rooms allocated currently.
**Key** – “unallocated” **Value -** ArrayList of Rooms unallocated currently.

Note : One Room can be occupied by one person only.