**Problem Statement:**

You are going to develop a Java application for hotels. A **Hotel** is actually a **Building** with some extra features. Each hotel has some features, such as the owner(s) and a couple of rooms. Each **Room**, has some features, including its own **Reservation** records. To implement and test this system, you must create **six Java classes, Person, Building, Reservation, Room, Hotel**, and **Tester**, as follows:

------------------------------------------------------------------------------------------------------------------------------------------------------------

**Class Person** (4 marks): Open the Java file **Person.java** that is provided, and complete it using the following specifications. Later, you must use this class in **Reservation** and **Hotel** classes.

* Every person has first and last names. (1 mark)
* Develop a constructor that gets two parameters, first and last names, and initializes the corresponding instance variables. (1 mark)
* Override the **toString** method, such that it returns first and last names with one space in between. (2 marks) ​

------------------------------------------------------------------------------------------------------------------------------------------------------------

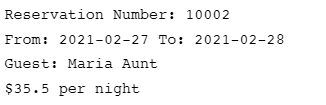
**Class Building** (4 marks): Open the Java file Building.java that is provided, and complete it using the following specifications:

* Every building has the year of built and number of floors. (1 mark) (Note: You must use the class Year, which is located in the java.time package for data type of instance variable year.)
* Develop the **default constructor** which initializes the year of built to the current year, and number of floors to 1. (2 marks) (Note: Current year must not be hard coded by a constant value, like 2022.)
* Develop another **constructor** with two parameters, the year of built and number of floors, which initializes the corresponding instance variables. (1 mark) ​

------------------------------------------------------------------------------------------------------------------------------------------------------------

**Class Reservation** (15 marks): Open the Java file **Reservation.java** that is provided. The purpose of this class is to keep the record of each reservation of a room. Later on, you must use this class inside the Room class. Complete this Java class using the following specifications:

* Every reservation instance object has (6 marks)
  + start date and end date (Note: Use Date class for the type of these two instance variables. Date is a class inside java.sql package.)
  + price per night
  + guest (Note: Type of this instance variable should be Person)
  + reservation number (Note: The first reservation number must be 100001. Therefore, it must be a class-level variable that keeps the last reservation number used.)
* Develop a **constructor** with four parameters, start date, end date, price, and guest. (1 mark) (Note: You must set the reservation number for the reservation in this constructor as well.)
* Develop getter methods for the start date, end date and the reservation number. (3 marks)
* Develop a method, **isAvailable**, which gets start date (inclusive) and end date (exclusive) and will check if the room is available during these dates or not. (3 marks)
* Override the **toString** method, such that it returns a string including the reservation number, start and end dates, guest, and price, with the format like example below. (2 marks)



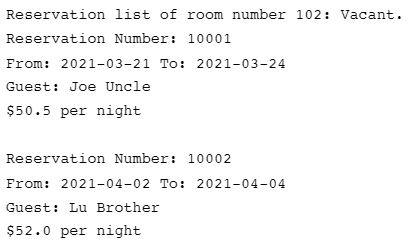
------------------------------------------------------------------------------------------------------------------------------------------------------------

**Class Room** (27 marks): Open the Java file Room.java that is provided. The purpose of this class is to keep the information of each room of a given hotel. Later on, you must use this class inside the Hotel class. Complete this Java class using the following specifications:

* Define an **enum**, named **RoomType**, with the four possible values below: (2 marks)

**SINGLE, DOUBLE, STUDIO, SUITE**

* Every room instance object has (8 marks)
  + Room type, price per night, room size, floor number, room number, description, vacancy status, and an ArrayList of all its reservations
* Develop a **constructor** with six parameters, room type, price, size, room number, floor, and description. Note that you must also initialize the vacancy status and the empty list of reservations. (3 marks)
* Develop getter methods for the vacancy status, room number, price, and list of the reservations. (2 marks)
* Develop a setter method for the price. (1 mark)
* Develop a method, **reserve**, which gets start date (inclusive), end date (exclusive), and guest, and will reserve the room. Note that the room availability should be checked in this method. If the room is available, then it reserves the room and returns the reservation number, and otherwise returns zero. (3 marks)
* Develop a method, **cancel**, that gets a reservation number and cancels it (2 marks)
* Develop two methods, **checkin** and **checkout**, to set the vacancy status of the room, accordingly. (2 marks)
* Develop a method, **status**, which returns a string including the list of all the reservations of the room, with a format like example below. (2 marks)



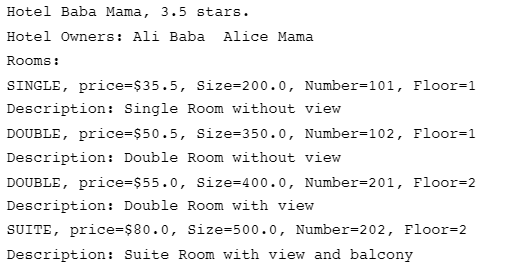
* Override the **toString** method, such that it returns a string including the room information, with the format like example below. The last part of the following example is room description.



------------------------------------------------------------------------------------------------------------------------------------------------------------

**Class Hotel** (30 marks): Open the Java file **Hotel.java** that is provided. Note that **every Hotel is a Building** with some extra features. You must consider this fact in the implementation of the class **Hotel**. Complete this Java class using the following specifications:

* Every hotel has Name, rate, one or more owners, and one or more rooms. (4 marks)
* Develop the **default constructor** for the class, which only calls the default constructor of its superclass. (1 mark)
* Develop the second **constructor** for the class, with five parameters, hotel name, list of the hotel owners, year of built, number of floors, and hotel rate, which initializes the corresponding instance variables. Note that the proper constructor of the superclass must be called in this constructor. Also, the empty list of the hotel rooms must be initialized. (4 marks)
* Develop getter methods for hotel name and owners. (2 marks)
* Develop a getter method, **getRooms**, which returns the list of the hotel rooms as an ArrayList of rooms. (1 mark)
* Develop a getter method, **getRoom**, with one parameter, room number, which returns the corresponding room as its return type, and otherwise null. (1 mark)
* Develop a method, **addowner**, which adds one owner to the hotel’s owners list. (1 mark)
* Develop a method, **addRoom**, which adds a new room to the hotel. (1 mark)
* Develop a method, **reserve**, with four parameters, room number, start date, end date, and guest. It will first find the room using one of the above methods. If it finds the room, then it simply calls the **reserve** method of that room, receives the reservation number and returns it. If the room is not found or if the room is not available for the period specified, it returns zero, which means the reservation was not successful. (4 marks)
* Develop a method, cancel, with one parameter, reservation number. It should find the room that has this reservation numbers. If it finds the room, then it will call the cancel method of that room with the reservation number given, and then returns true. Otherwise, it returns false, which means the cancellation was not successful. (3 marks)
* Override the toString method, such that it returns a string including the hotel information, followed by the rooms’ information, like below: (3 marks) ​



* Develop the **Standard Java Documentation** for **all the public methods** as well as one brief explanation for the **Hotel** class at the top of the class. Note that your documentation must follow the correct syntax of Java Documentation and not just simple comments. (5 marks)

**Class Tester** (20 marks): For this part, you have two options:

**Option 1**: You can complete the tester class, **Tester.java**, that is provided. You must read every comment and provide the required code to fulfill it. You can also get help and idea from the sample execution outputs provided below. (20 marks)

**Option 2**: You can develop your own tester class, **Tester.java**. If you select this option, your tester class must generate outputs similar to the sample execution outputs provided below. This means, the main method of your tester class must:

* Create a hotel, with at least two owners and four rooms (6 marks)
* Show the hotel and its rooms’ info like the sample outputs. (4 marks)
* Perform three successful and one unsuccessful room reservations and show the results (6 marks)
* Change the current price of a room (1 mark)
* Cancel one existing reservation (1 mark)
* Show the statuses of all the rooms (2 marks)

Expected output:

