BIT106 Programing in Java I

Assignment 2 **Semester 3, 2020**

Release Date:	
Due date:	
Value:	15%
Assessment Mode:	Individual Assessment
Late Penalty	5 marks per day

The assignment assesses the following learning outcomes:

CLO1: use the fundamental control structures and data types to write programs in the Java language
CLO2: demonstrate the use of objects and classes, both standard library classes and user defined classes
CLO3: apply the structured, and object-oriented approach to solve computing problems using Java

Important Notes

- You must submit this assignment as a single word document to turnitin.com and hardcopy to the School of Information and Communication Technology.
- You must work on this assignment on your own. Any **two or more** programs which are found to be similar will receive **zero (0)** marks.
- For any queries/clarifications, revert as soon as possible. Any queries made close to the deadline may not be possible to entertain.

Overview

Affirvat Hostels provides affordable accommodation to students pursuing coaching for various competitive examination in metropolitan cities. It requires a software application which can be used by its staff to manage various operations. Some of the main operations are as follows:

- Book a room
- Vacate a room
- Display available rooms
- Search room details
- Display summary information about rooms

You are required to develop an application using JAVA that will allow the hostel staff to interactively perform various tasks as stated above.

Assignment Requirements:

You are required to write the following three classes:

- `Room`: A class, defines a simple object type representing a room.
- `Hostel`: A class, defines objects which are containers of `Room` objects.
- `HostelMain`: A driver class that creates one `Hostel` object and allows the various methods of `Hostel` to be called. This class will provide interactive application interface using the keyboard and the screen to the hostel staff. This is only providing an interface and will not do calculations itself but will immediately pass user inputs as arguments to methods of `Hostel` class.

NOTE: The final application will only execute correctly when all three classes have been defined completely and correctly but don't wait until you have completely written all three before you start compiling and testing your code. It is recommended that you save all three source code files in the same directory on your file system and compile and test each class as you develop it using small separate programs to create and test objects of each class.

The files

The files you will require are:

Room.java

This file defines a class of `Room` objects. The objects have the following instance variables:

- number of beds in the room, of type integer;
- guest's name, of type String;
- booking status of a room – if the room has been booked, the status is 'true', otherwise it is 'false';
- room tariff, i.e. cost of using the room for one night, of type double;

The methods of class `Room` should include:

- A default constructor. This constructor should initialise a Room object with the number of beds as two, the guest's name as "Nobody", the booking status as false, and the room cost to 100.00. This is the default state of a Room object.

- A Setter method that accepts one argument which is used to set the number of beds. It must ensure that the number of beds stored in the Room object remains in the range 1-4 inclusive.
- A Setter method for the room tariff. This method accepts one argument representing a new tariff value. It must ensure the tariff is not negative.
- A method called `bookRoom` which accepts a String argument representing a guest's name. It sets the booking state to true and assigns the parameter value to the guest's name variable.
- A method called `vacateRoom` which sets the booking state to false and sets the guest's name variable back to "Nobody".
- A getter method for each of the class members which are number of beds, the tariff and the guest's name - these return the appropriate value.
- A boolean method called `isBooked` which returns the booking status.
- A `'toString'` method which return a single String containing the details of a room with format as described below:

  `Room with <numOfBeds> beds, tariff <roomTariff>, and guest named <guestName>.`

  or

  `Room with <numOfBeds> beds, tariff <roomTariff>, and is vacant.`

  Example:

  `Room with 2 beds, tariff 100.00, and guest named James Bond.`

  or

  `Room with 2 beds, tariff 100.00, and is vacant.`

It is recommended that once you have written the `Room` class, you create a program to test it. The testing program should be placed in the same working directory as the `Room` class and be used to create one or two `Room` objects and call some of the `Room` methods. Compile the `Room` class and compile and run the test program to check your work.

Hostel.java

This file declares a class which maintains a collection of `Room` objects. It will contain methods which enable the collection to show the appropriate behaviour as required by the menu. This file should be saved into the same working directory as `Room.java`.

The `Hostel` class should declare an array of Room objects; no additional attribute is allowed.

The `Hostel` class must also contain methods which allow the collection of rooms to be managed. These methods should include:

- A constructor which accepts an integer which is used to set the size of the `Room` array. If the integer value passed in is invalid, then an array of Room objects of size 50 is to be created. If the parametric integer is valid, that is between 20 and 100, inclusive, then a `Room` array of the specific parametric value will be created. Next you need to perform some initialisation tasks for the rooms as described below. Each task can be defined as private method, and the constructor will then invoke these methods to complete the task:
  - ✓ First task is to traverse the array and instantiate a default Room object referenced by each array cell. After each Room has been instantiated, we will assume that the array index will represent the room number in the hostel. For example, room number 2 will be in array cell with index 2, room numbered 5 will be in array cell with index 5, etc.
  - ✓ Second task is to traverse the array and set the room tariff of all the even numbered rooms to $150.00, except room numbered 0, which is set to $1500.00 as it is the penthouse suite.

- ✓ Third task is to set the number of beds to 1 for the last 5 rooms, and set the number of beds to 4 for rooms 1 through 5 inclusive.

- A method named `getRoom` which accepts an integer parameter representing a room number and returns a reference to the Room object in that cell of the array. If the parametric integer is illegal, a null reference should be returned.
- A method named `numOfBookedRooms` which does not accept any parameter, and returns the number of rooms which are booked.
- A method named `numOfVacantRooms` which does not accept any parameter, and returns the number of rooms which are not booked.
- A method named `totalTariff` which does not accept any parameter, and returns the total value of all the tariffs of all the booked rooms. This simulates one day's income for the hostel.
- A method named `getAvailableRooms` which accepts an integer representing a number of guests which need a room. This method should return a String in which there is a list of all the vacant rooms which have enough beds for the prospective guests.
- A method named `findGuestRoomNumber` which accepts a String representing a guest's name and searches through all the rooms looking for the first guest whose name is the same as the parametric name. The method should return the number of the room when a match is found. If the name cannot be found, the method should return -1.

When you have written the `Hostel` class - test it by creating a `Hostel` object and invoking the methods from a Java program.

HostelMain.java

The aim of this class is to provide a user-interface for a modest application which uses a `Hostel` container class and should be saved in the same working directory as the previous files. It is recommended that this user-interface be written as a 'console' application using the normal screen and keyboard to interact with a user via a simple text-based menu.

The user-interface should create a single `Hostel` object and provide a menu of choices to the user with the following choices:

1 See available rooms for 'n' guests

The operator enters the number of guests needing accommodation. This value should then be passed to the `getAvailableRooms` method of the `Hostel` object, the returned String captured and displayed. This tells the operator which rooms can be booked.

2 Book a room

The operator enters name of the guest, then fetches the Room object of an appropriate empty room (using the 'getRoom' method) and books it with the guest's name.

3 Vacate a room

The operator enters the number of the room to be vacated, the Room object with that number is obtained using 'getRoom' and vacated. If the room is not booked, display an appropriate message.

4 Find which room a guest is in.

The operator enters a guest's name and this is passed to the `'findGuestRoomNumber'` method and the room number is displayed. If no such guest is found, display an appropriate message.

5 Print a report

Display

- the number of booked rooms
- the number of empty rooms
- the total tariff of all booked rooms

6 Other options

You may add any other options you like to this list and you will be given credit for those which are implemented correctly.

7 Quit the program.

Each time the user selects one of the previous options, and the program does that task, the menu should be presented again. If they choose to quit, the program should end.

Documentation

Document all classes and its methods with the purpose, and your name and date. Use meaningful variable names and comments.
Your sample output should cover all the requirements.
Refer to the marking scheme for further details. You must submit the marking scheme with your assignment submission.

What to Submit (Turnitin)

Submit a single word document with all your source code in courier size 10 to turintin.

Turnitin.com (Assignment 2) *class ID : 26616679* *Enrolment password: javasem32020*

Late submissions will be penalized 5 marks per working day.

SUBMIT ON LMS:

1. **Word file that includes the following (in order).**
   - Assignment Cover Sheet
   - Program source code, formatted with Courier New size 10
   - Sample Output
   - Turnitin report
   - Assignment Marking Scheme
2. **Export your NetBeans project into a zipped file.**