ASSIGNMENT 4

COMP-202, Fall 2017, All Sections

Due: Wednesday, November 22^{nd}, 11:59pm

Please read the entire PDF before starting. You must do this assignment individually.

Question 1:	100 points
	100 points total

It is very important that you follow the directions as closely as possible. The directions, while perhaps tedious, are designed to make it as easy as possible for the TAs to mark the assignments by letting them run your assignment, in some cases through automated tests. While these tests will never be used to determine your entire grade, they speed up the process significantly, which allows the TAs to provide better feedback and not waste time on administrative details. Plus, if the TA is in a good mood while he or she is grading, then that increases the chance of them giving out partial marks. :)

Up to 30% can be removed for bad indentation of your code as well as omitting comments, coding structure, or missing files. Marks will be removed as well if the class and method names are not respected.

To get full marks, you must:

- Follow all directions below
- Make sure that your code compiles
  - Non-compiling code will receive a very low mark
- Write your name and student ID as a comment in all .java files you hand in
- Indent your code properly
- Name your variables appropriately
  - The purpose of each variable should be obvious from the name
- Comment your work
  - A comment every line is not needed, but there should be enough comments to fully understand your program

Part 1 (0 points): Warm-up

As usual, do **NOT** *submit this part, as it will not be graded.*

Warm-up Question 1 (0 points)
Write a class describing a `Cat` object. A cat has the following `attributes`: a name (String), a breed (String), an age (int) and a mood (String). The mood of a cat can be one of the following: `sleepy, hungry, angry, happy, crazy`. The cat `constructor` takes as input a String and sets that value to be the breed. The `Cat` class also contains a method called `talk()`. This method takes no input and returns nothing. Depending on the mood of the cat, it prints something different. If the cat's mood is `sleepy`, it prints *meow*. If the mood is `hungry`, it prints *RAWR!*. If the cat is `angry`, it prints *hsssss*. If the cat is `happy` it prints *purrrr*. If the cat is `crazy`, it prints a String of 10 gibberish characters (e.g. raseagafqa).

The cat `attributes` are all **private**. Each one has a corresponding **public** method called `getAttributeName()` (ie: `getName(), getMood()`, etc.) which returns the value of the `attribute`. All but the `breed` also have a **public** method called `setAttributeName()` which takes as input a value of the type of the attribute and sets the attribute to that value. Be sure that only valid mood sets are permitted. (ie, a cat's mood can only be one of five things). There is no setBreed() method because the breed of a cat is set at birth and cannot change.

Test your class in another file which contains only a main method. Test all methods to make sure they work as expected.

Warm-up Question 2 (0 points)
Using the `Cat` type defined in the previous question, create a `Cat[]` of size 5. Create 5 `Cat` objects and put them all into the array. Then use a loop to have all the `Cat` objects `meow`.

Warm-up Question 3 (0 points)
Write a class `Vector`. A `Vector` should consist of three `private` properties of type double: x, y, and z. You should add to your class a constructor which takes as input 3 doubles. These doubles should be assigned to x, y, and z. You should then write methods `getX()`, `getY()`, `getZ()`, `setX()`, `setY()`, and `setZ()` which allow you to get and set the values of the vector. Should this method be static or non-static?

Warm-up Question 4 (0 points)
Add to your `Vector` class a method `calculateMagnitude` which returns a double representing the magnitude of the vector. Should this method be static or non-static? The magnitude can be computed by taking:

$$magnitude = \sqrt{x^2 + y^2 + z^2}$$

Warm-up Question 5 (0 points)
Write a method `scalarMultiply` which takes as input a `double[]`, and a `double scale`, and returns `void`. The method should modify the input array by multiplying each value in the array by `scale`. Should this method be static or non-static?

Warm-up Question 6 (0 points)
Write a method `deleteElement` which takes as input an `int[]` and an `int target` and deletes all occurrences of `target` from the array. By "delete" we mean create a new array (of smaller size) which has the same values as the old array but without any occurrences of `target`. The method should return the new `int[]`. Question to consider: Why is it that we have to return an array and can't simply change the input parameter array like in the previous question? Should these methods be static or non-static?

Warm-up Question 7 (0 points)
Write the same method, except this time it should take as input a `String[]` and a `String`. What is different about this than the previous method? (Hint: Remember that `String` is a reference type.)

Part 2

The questions in this part of the assignment will be graded.

Question 1: Booking System (100 points)

For this question, you will write several classes, and then create and use instances of those classes in order to simulate a hotel booking system. Note that in addition to the required methods below, you are free to add as many other `private` methods as you want (no additional `public` method is allowed).

(a) (20 points) Write a class `Room`. A Room has the following *private* attributes:

- A `String` type
- A `double` price
- A `boolean` availability

The Room class also contains the following *public methods*:

- A constructor that takes as input the type of the room and uses it to initialize the attributes. Note that there are only 3 type of Rooms supported by the program: *double*, *queen*, and *king*. If the input is not one of these types, then the constructor should throw an IllegalArgumentException explaining that no room of such type can be created. The price of the room is based on its type as follows: \$90 for a double, \$110 for a queen, \$150 for a king. The constructor should set the availability for a new room to be true.
- `getType` which returns the type of the room.
- `getPrice` which returns the price of the room.
- `getAvailability` which returns the availability of the room.
- `changeAvailability` which takes no input and sets the value stored in the availability attribute to be the opposite of the one currently there.
- A `findAvailableRoom` method which takes as input an array of Rooms as well as a String indicating the room type. The method should return the first available room in the array of the indicated type. If no such room exists (either because all rooms of said type are occupied, or because no room of such type is in the array), the method returns null.

(b) (10 points) Write a class `Reservation`. A Reservation has the following *private* attributes:

- A `String` name
- A `Room` roomReserved

The Reservation class also contains the following *public methods*:

- A constructor that takes as input a Room and the name under which the reservation is made. The constructor must use its inputs to initialize the attributes.
- `getName` which returns the name under which the reservation was made.
- `getRoom` which returns the Room that has been reserved.

(c) (40 points) Write a class `Hotel`. This class has three *private* attributes:

- A `String` name
- An array of `Room`
- An array of `Reservation`

It also has the following methods:

- A `public` constructor that takes as input the name of the hotel, and an array of rooms. It initializes the `Room` array and *copies* the Room references from the input array to the Hotel array.
- A `private` method `addReservation` that takes a Reservation as input and add it to the array of Reservation of the hotel.
- A `private` method `removeReservation` that takes as input a name and a type of room. If no reservation has been made under the given name for the given type of room, then the method should throw a `NoSuchElementException` with the appropriate message. If a matching reservation is found, then the method should remove such reservation from the array of reservations of the hotel. Note that you need to import `java.util.NoSuchElementException` in order to be able to use it.
- A `public` method `createReservation` that takes as input a name and the type of the room that should be reserved. The method should use `findAvailableRoom` (from the `Room` class) to verify whether a room of the given type is available in the hotel. If not, the method should print a message informing the user that the hotel has no available rooms of the requested type. Otherwise, the method should: create a reservation for such room under the given name, change the availability of the room, and add the reservation to the array of reservations of the hotel using the `addReservation` method. In this case, the method should also print a message letting the user know that the reservation was successfully completed.
- A `public` method `cancelReservation` that takes as input the name under which the reservation was made and the room type reserved. This method should use the `removeReservation` method to cancel the reservation. If no matching reservation was found, the method should inform the user that no reservation under such name has been made for the given type of room. Otherwise, the method should print a message letting the user know that the operation was successful. To receive full marks, a try/catch block should be used.
- A `public` method `printInvoice` that takes as input a name and prints out a message letting the user know how much the given person owes the hotel based on *all the reservations* made under such name.
- The `toString` method that returns a String containing the name of the hotel as well as how many *available* rooms of each type the hotel has.

(d) (30 points) In the provided class `BookingSystem` write a main method.
Here, you will write a main method that creates an hotel where the name of the hotel is chosen by the user via a Scanner object. The program then creates an array of rooms at random and uses it, together with the hotel name, to create a new hotel.

In order to facilitate generating your Room array, we have provided methods to generate a random number of rooms, and to select the room types randomly from a provided array.

Then, the program should print on the screen a menu from which the user can chose one of the following options:

1. Make a reservation
2. Cancel a reservation
3. See an invoice
4. See the hotel info
5. Exit the booking system

Depending on the option chosen, you should use Scanner to obtain from the user the information needed (if any) to carry forward the operation selected. After each operation, the main menu should be displayed again, unless the user has chosen to exit the booking system. In such case, the program should end. If the user enters a wrong option, then the main menu should also be displayed again. Note that you can assume that the user will always be entering an integer. Sample interaction and output is shown below.

First example:

```
> run BookingSystem
Welcome to the COMP 202 booking system
Please enter the name of the hotel you'd like to book
 Grand Budapest Hotel

**********************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
**********************************************************************

 4

Here is the hotel info
Hotel name: Grand Budapest Hotel
Available Rooms: 10 double, 8 queen, 15 king.


**********************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
**********************************************************************

 1
Please enter your name: Gustave
What type of room would you like to reserve? king
You have successfully reserved a king room under the name of Gustave. We look forward having you at Grand Budapest Hotel!

**********************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
**********************************************************************

 1
Please enter your name: Gustave
What type of room would you like to reserve? king
You have successfully reserved a king room under the name of Gustave. We look forward having you at Grand Budapest Hotel!

**********************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
**********************************************************************

 1
Please enter your name: Gustave
What type of room would you like to reserve? queen
```

```
You have successfully reserved a queen room under the name of Gustave. We look forward having you at Grand Budapest Hotel!

**********************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
**********************************************************************

 4

Here is the hotel info
Hotel name: Grand Budapest Hotel
Available Rooms: 10 double, 7 queen, 13 king.


**********************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
**********************************************************************

 3
Please enter your name: Gustave
Gustave's invoice is of $410.0

**********************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
**********************************************************************

 5

It was a pleasure doing business with you!
>
```

Second example:

```
> run BookingSystem
Welcome to the COMP 202 booking system
Please enter the name of the hotel you'd like to book
 Grand Budapest Hotel

************************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
************************************************************************

 4

Here is the hotel info
Hotel name: Grand Budapest Hotel
Available Rooms: 2 double, 1 queen, 0 king.


************************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
************************************************************************

 1
Please enter your name:  Gustave
What type of room would you like to reserve?  king
Sorry Gustave, we have no available rooms of the desired type.

************************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
************************************************************************

 1
Please enter your name:  Gustave
What type of room would you like to reserve?  queen
You have successfully reserved a queen room under the name of Gustave. We look forward having you at Grand Budapest Hotel!

************************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
************************************************************************

 1
Please enter your name:  Gustave
What type of room would you like to reserve?  double
You have successfully reserved a double room under the name of Gustave. We look forward having you at Grand Budapest Hotel!
```

```
************************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
************************************************************************

 2
Please enter the name you used to make the reservation  Gustave
What type of room did you reserve?  king
There's no reservation for a king room under the name of Gustave.

************************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
************************************************************************

 2
Please enter the name you used to make the reservation  Gustave
What type of room did you reserve?  double
Gustave, your reservation for a double room has been successfully cancelled.

************************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
************************************************************************

 3
Please enter your name:  Zero
No reservations have been made at this name.

************************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
************************************************************************

 3
Please enter your name:  Gustave
Gustave's invoice is of $110.0

************************************************************************
Welcome to Grand Budapest Hotel. Chose one of the following options:
1) Make a reservation
2) Cancel a reservation
3) See an invoice
4) See hotel info
5) Exit the Booking System
************************************************************************

 5

It was a pleasure doing business with you!
> |
```

What To Submit

Please put all your files in a folder called Assignment4_ID, where 'ID' should be replaced by your McGill ID number. Zip the folder (please DO NOT rar it) and submit it in MyCourses. Inside your zipped folder there must be the files listed below. **Do not submit any other files, especially .class files**.

`BookingSystem.java`
`Hotel.java`
`Reservation.java`
`Room.java`
`Confession.txt` (optional) In this file, you can tell the TA about any issues you ran into doing this assignment. If you point out an error that you know occurs in your problem, it may lead the TA to give you more partial credit. On the other hand, it also may lead the TA to notice something that otherwise they would not.