TCSS 305 Programming Practicum
Assignment 4: Rentz – Vehicle Rental System
Value: 13% of the course grade
Due: Friday, 28 February 2019, 23:59:00

Program Description:

Assignment 0, Assignment 1, 2, 3, 4 aim to develop Rentz, a vehicle rental system. You will incrementally build your code for each assignment, i.e., your Assignment 2 will be built using the code that you submitted for Assignment 1. Make sure you finish every assignment on time, so that your future submissions remain unaffected. Assignment 4 aims to develop a full-fledged vehicle rental system Rentz.

Rentz is operated by a rental agent. The rental agent should be a registered user. The rental agent can 1) Register New Users; 2) Login to the rental system; 3) Rent a vehicle for a registered user; 4) Drop-off a vehicle for a registered user; 5) Exit from the system. In Assignment 1-3, you have already created and tested the registration and login functionalities. Assignment 3-4 will focus on the remaining tasks.

Using Previous Code: Rename you project to "username-rentz-4" and share it to the repository following steps 1-3 under "Using Previous Code" given in "hw1.pdf".

Implementation Guidelines:

Package `model`: You will need to add the `Bill` class and make changes to the `RentalManager` class. *You can add class private/public fields, methods, apart from the ones mentioned in the guidelines if necessary.* Unnecessary class members will be penalized.

`Bill`:

Field	**Description**		
`myBillID`	A unique integer value		
`myPrimaryUser`	A User object		
`myVehicle`	A vehicle object		
`myNumDays`	An integer representing the number of days the vehicle is rented		
`myBillAmount`	Total bill amount (BigDecimal Value)		
Method	**Parameters**	**Return Type**	**Description**
`Parameterized Constructor`	All fields except `myBillAmount`		Initializes all fields. `myBillAmount` should be initialized to 0
`computeAndPrintAmount`			Computes and prints `myBillAmount` as shown in execution output US Number formatting should be used, while printing cost Total Rental is calculated based on the

			following entities • Rental amount = numOfDays * vehicle rental cost per day • Add Insurance which is 1% of rental amount • If VIP status, apply discount of 1% rental amount • Add Tax, which is 10% of the total amount calculated after applying insurance and VIP discount to the rental amount.

`RentalManager`: Update Rental Manger based on the following guidelines

Field	**Description**		
`myBills`	HashMap which saves with Key= `myBillID`, Value=Bill Object The `myBillID` starts from 1 and is sequentially generated for every Bill by the system		
Method	**Parameters**	**Return Type**	**Description**
`Parameterized Constructor`	`Registration` Object		`myBills` is initialized to a new HashMap object
`isRentable`	`int theVehicleID`	boolean	Returns true if the vehicle is a part of the inventory, i.e., `myVehicleList` and if it is not already rented

`rent`	`int theVehicleID`
	
	`String theUserName`
	
	`int theNumDays`
	
	`int theBillID`
			• Appropriate parameters should be checked for null values/illegal arguments. `NullPointerException` (implicitly) and `IllegalArgumentException` (explicitly) should be thrown when necessary.
			• Checks if `theVehicleID` is rentable by calling `isRentable`
			• Checks if `theUserName` is a registered user
			• If the above 2 conditions are true
			1. vehicle's rental availability is set `false`
			2. create a `Bill` Object
			3. call `computeAndPrintAmount` of the Bill object
			4. Add the `Bill` object to `myBills`
			• Returns `true` if successful and `false` if not

`drop`	`int theVehicleID`		• Checks if `theVehicleID` is valid and is not rentable by calling `isRentable`

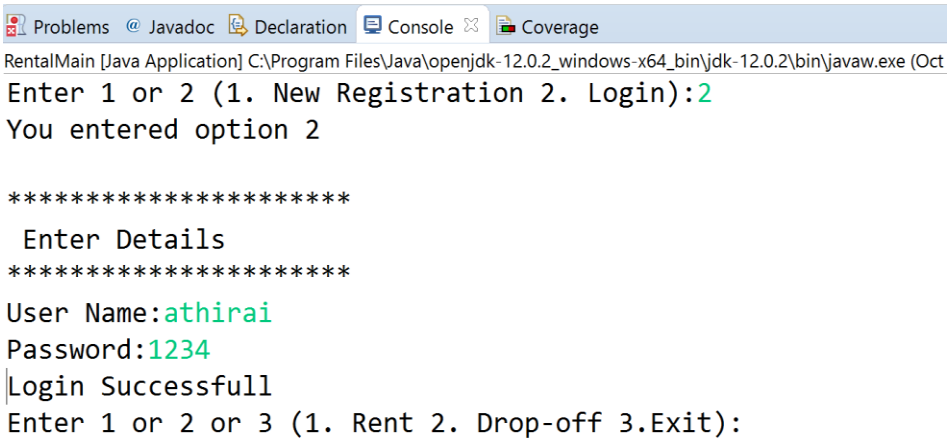
			• If the above condition is satisfied make the rental state of the vehicle to be `true` • Returns `true` if drop-off successful else `false`
`printOptions`			The exact output is shown in the execution output: Takes in input for (1. Rent 2.Drop-off 3.Exit) Rent: 1. Lists the vehicles available for renting 2. Generates `myBillID` 3. Gets the VehicleID, UserName, NumDays as input form console 4. Calls `rent` method. Repeats Step 3 until `rent` method returns true Drop-off: 1. Gets VehicleID for drop-off 2. Calls `drop` method. Repeat Step 1 until `drop` method returns true Exit: 1. Returns from the `printOptions` function The whole iteration should be repeated based on whether the agent wants to quit or not using a `boolean` variable.
`clearLists`			Clears `myVehicleList` and `myBills`

Notes:

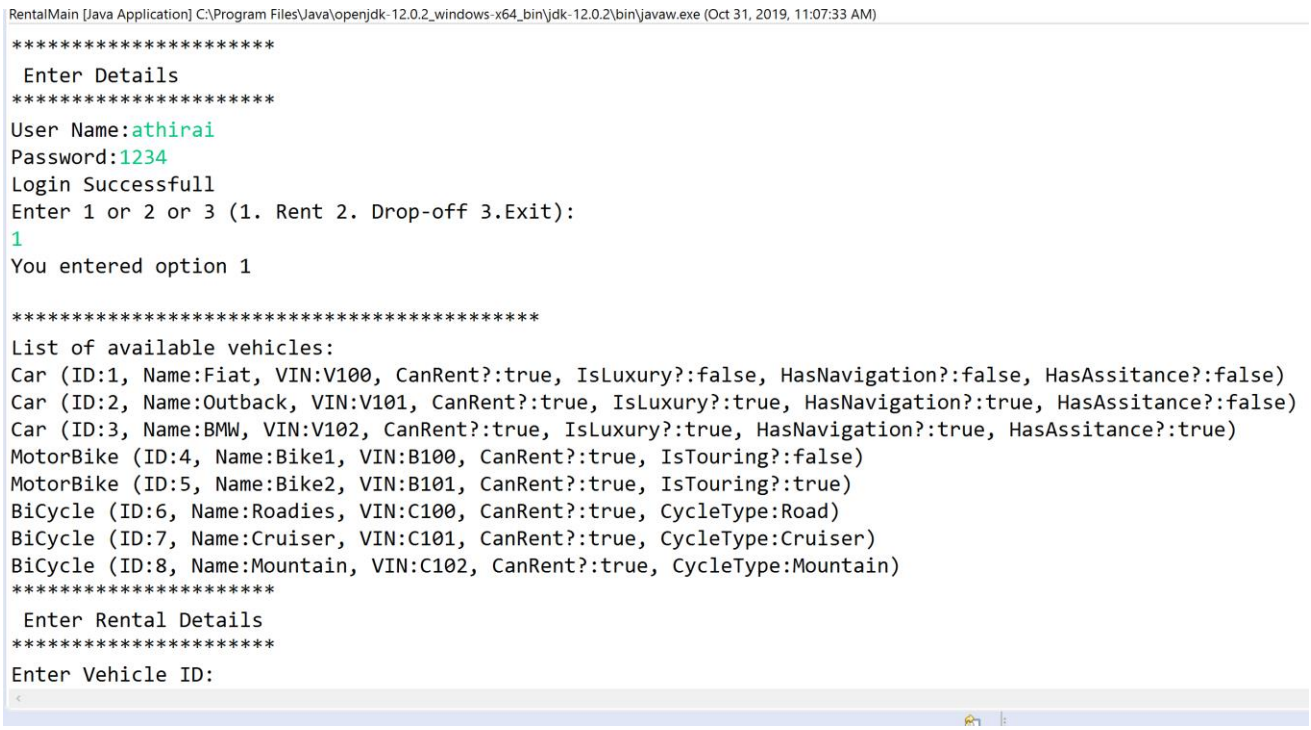
- You can use utility methods to reduce overloading `printOptions` method
- You are allowed to create a public static final field for `Scanner` in `Registration` or any of the classes. You should use the same scanner object for all classes. You may want to close the Scanner object in `RentalMain` class after all other operations are completed.
- You are allowed to create a public static final field for `Number Format` in your Vehicle or RentalManager class. You should use the same Number Format instance in all the classes.
- Your code should printout the contents exactly as shown in the execution output.

Expected Output: When you run the `RentalMain` class this should be the expected output (Exact formatting should be followed)

1. After login the 3 options should be displayed as shown below:



2. After selecting Option 1, the list of **available** vehicles should be displayed and it should ask for VehicleID, UserName, NumDays



3. After entering VehicleID, UserName, NumDays, if the inputs are valid, the Bill is generated and displayed as shown below. It also asks, if you want to continue?

```
**********************
Enter Vehicle ID:1
Enter User Name:athirai
Enter NumDays to Rent:2

**********************
 Rental Bill Summary
**********************
User Name: athirai
----Vehicle Information----
VehicleName Fiat
VehicleID 1
VehicleType V100
VIN V100
----Cost Information----
RentalPerDay:
Cost per Day: $30.00
No.of Rental days: 2
Total Amount: $60.00
Insurance: $0.60
VIPDiscount: -$0.60
Tax: $6.00
Total Rent: $66.00
Rent Successfull
**********************
 Do you want to continue?
```

4. When the agent enters “true” for “Do you want to continue?”, it displays the 3 options again. When I select option 1, the list of available vehicles is displayed. (The rented vehicle should not be in the list)

```
 Do you want to continue?true
Enter 1 or 2 or 3 (1. Rent 2. Drop-off 3.Exit):
1
You entered option 1

*****************************************
List of available vehicles:
Car (ID:2, Name:Outback, VIN:V101, CanRent?:true, IsLuxury?:true, HasNavigation?:true, HasAssitance?:false)
Car (ID:3, Name:BMW, VIN:V102, CanRent?:true, IsLuxury?:true, HasNavigation?:true, HasAssitance?:true)
MotorBike (ID:4, Name:Bike1, VIN:B100, CanRent?:true, IsTouring?:false)
MotorBike (ID:5, Name:Bike2, VIN:B101, CanRent?:true, IsTouring?:true)
BiCycle (ID:6, Name:Roadies, VIN:C100, CanRent?:true, CycleType:Road)
BiCycle (ID:7, Name:Cruiser, VIN:C101, CanRent?:true, CycleType:Cruiser)
BiCycle (ID:8, Name:Mountain, VIN:C102, CanRent?:true, CycleType:Mountain)
**********************
 Enter Rental Details
**********************
Enter Vehicle ID:
```

5. When the agent enters a non-existing/unavailable vehicleID or an un-registered user, system should prompt to re-enter details until the correct information is entered. All details (vehicleID, username, numdays) should be entered at once, then the validity check should be performed.

```
 Enter Rental Details
**********************
Enter Vehicle ID:9
Enter User Name:athirai
Enter NumDays to Rent:3
Vehicle not rentable
**********************
 Enter Rental Details
**********************
Enter Vehicle ID:2
Enter User Name:305RentzUser
Enter NumDays to Rent:3
User does not exists, enter different user name:
**********************
 Enter Rental Details
**********************
Enter Vehicle ID:2
Enter User Name:athirai
Enter NumDays to Rent:3

**********************
 Rental Bill Summary
**********************
User Name: athirai
----Vehicle Information----
VehicleName Outback
VehicleID 2
```

6. When the agent enters option 2, the VehicleID for drop-off should be prompted

```
Rent Successfull
**********************
 Do you want to continue?true
Enter 1 or 2 or 3 (1. Rent 2. Drop-off 3.Exit):
2
You entered option 2

**********************************************
**********************
 Enter Drop-off Details
**********************
Enter Drop-off Vehicle ID:
```

7. When the Drop-off is successful, the following should be printed

```
**********************
 Enter Drop-off Details
**********************
Enter Drop-off Vehicle ID:1
Drop-off Successfull
**********************
 Do you want to continue?
```

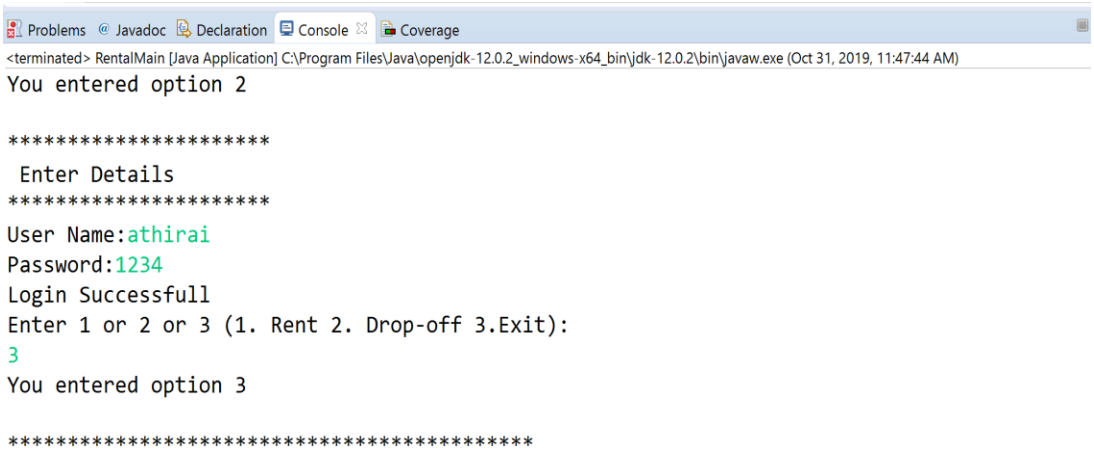
8. When a non-existing/non-rented VehicleID is enter the system should prompt to re-enter VehicleID until correct information is collected.

```
Problems  Javadoc  Declaration  Console  Coverage
RentalMain [Java Application] C:\Program Files\Java\openjdk-12.0.2_windows-x64_bin\jdk-12.0.2\bin\javaw.e
You entered option 2

**********************************************
**********************
 Enter Drop-off Details
**********************
Enter Drop-off Vehicle ID:4
Vehicle is not rented already
Enter Drop-off Vehicle ID:10
Vehicle does not exists
Enter Drop-off Vehicle ID:1
Drop-off Successfull
**********************
 Do you want to continue?
```

9. When the agent does not want to continue, it enters “false” and the program terminates

```
Problems  Javadoc  Declaration  Console  Coverage
<terminated> RentalMain [Java Application] C:\Program Files\Java\openjdk-12.0.2_windows-x64_bin\jdk-12.0.2\bin\javaw.exe (Oct 31, 2019, 11:43:59 AM)

**********************************************
**********************
 Enter Drop-off Details
**********************
Enter Drop-off Vehicle ID:4
Vehicle is not rented already
Enter Drop-off Vehicle ID:10
Vehicle does not exists
Enter Drop-off Vehicle ID:1
Drop-off Successfull
**********************
 Do you want to continue?false
```

10. The program also terminates when option 3 is selected



Extra Credit (Upto 10 points):

You can choose to implement a maximum of 2 among the following options for extra credit.

1. The Bills generated should be saved as a .txt file under the `resources` folder. A separate function for writing to a file should be added to the Bill class [5 points]
2. Apart from the 3 options for (Rent, Drop-off and Exit) [5 points]
    - [2.5 points] Add an option for displaying vehicles (both available and non-available) based on Type, the system should prompt for Vehicle Type and vehicles of the particular type should only be displayed
    - [2.5 points] Add an option for displaying Rental History, the system should prompt for vehicleID and the rental history: user, numDays, rental amount should be displayed. The rental history should include all previous rentals for the vehicleID
3. Apart from the 3 options for (Rent, Drop-off and Exit) [5 points]
    - Add an option for displaying vehicles (both available and non-available), sorted based on their rental amount per day

Submitting your executive summary:

The executive summary template will be provided to you in canvas. Download the file. Rename the executive summary "`executivesummary-username-rentz-4`", where `username` is your UWNetID. Fill and upload the *executive summary* to canvas as the same WORD format. The grader will grade your submission based on the executive summary as well as the project you submit to the SVN repository. If you do not submit either of them, you will receive no grades.

Grading Rubric (Total 100 points (110 possible)):

Task	Max Score Possible
Executive Summary -Submission on canvas with correct version number	2
Source Code – Submission to SVN	2
Bill Class – Appropriate fields and modifiers [code]	2
Bill Class – correct implementation of constructor [code]	2
Bill Class – correct implementation of computeAndPrintAmount [code]	5
RentalManager – correct fields with proper modifiers created	1
RentalManager – constructor implementation	1
RentalManager – printOptions implementation [code]	7
RentalManager – rent implementation [code]	5
RentalManager – isRentable implementation [code]	4
RentalManager – drop implementation [code]	4
RentalManager – clearLists implementation [code]	1
Execution Output	
New User Registration (+ re-enter name if already existing)	5
User Login (+ invalid login credentials)	5
Print (Rent, Drop-off, Exit) options after login	2
Rent: Display only rentable vehicles in correct format	5
Rent: Prompt vehicleID, userName, numDays	2
Rent: Re-enter vehicleID, userName, numDays , if invalid	2
Rent: Bill Generation in correct format	10
Rent: Rent Successful, Do you want to continue?	5
Rent: Rent again for the second time. The rented vehicle should not be displayed	5
Drop: prompt to enter vehicleID	2
Drop: prompt to re-enter vehicleID if invalid	2
Drop: drop-off successful	2
Rent: Rent the previously dropped vehicle	2
Terminate: Select “False” for “Do you want to continue”	2
Terminate: Select Option3	2
Correct Project Name and Executive summary name	1
No Errors and Warnings (except reasonable ones which should be clearly explained in executive summary)	5
Proper Java docs and Header comments	5
Extra Credit Task1	5
Extra Credit Task 2	5
Total (110 possible)	100