CMPP269 Assignment #3 – R&R E-Bikes

Instructions

Rikki and RJ have decided to start a small E-Bike rental business in a nearby resort. They have decided that no advance reservations will be taken and that rentals can only be done in person, on a first-come first-served basis. Your team has been hired to develop the application program to handle rentals. Some design and programming work has already been done, including complete source code for a Bike class, which you are to use in your application. Further details about this are given on the next page.

Your application should use the provided comma-separated value file, ***bikes.csv***, to both retrieve and store the necessary information about bikes. The format of this file is given below.

> <bike id>,<battery charge>,<in use?>,<rental start time>,<user name>
>
> For example here are two bikes, one not in use and the other currently rented:

```
Fast1,0.70,false,0,
Trax,1.00,true,1648757715117,r.lee@hotmail.com
```

Your application should provide the following basic functionality:

- On start up, read in the most recent bike information from the data file and put it into an array or ArrayList of Bike objects
- Continually input and process user selections until user chooses to exit program
  - Display the list of all bikes with their current information
  - Display menu options (see below), get the user's selection, and perform action(s) appropriate for the selected option. The menu options are:
    1. Rent a bike
    2. Return a bike
    3. Charge battery
    4. Exit program
  - Display feedback indicating whether or not the selected option was performed
- On exit, write the current bike information from your array or ArrayList of Bike objects back to the data file

Note: You DO NOT need to write to the data file each time a menu selection is processed.

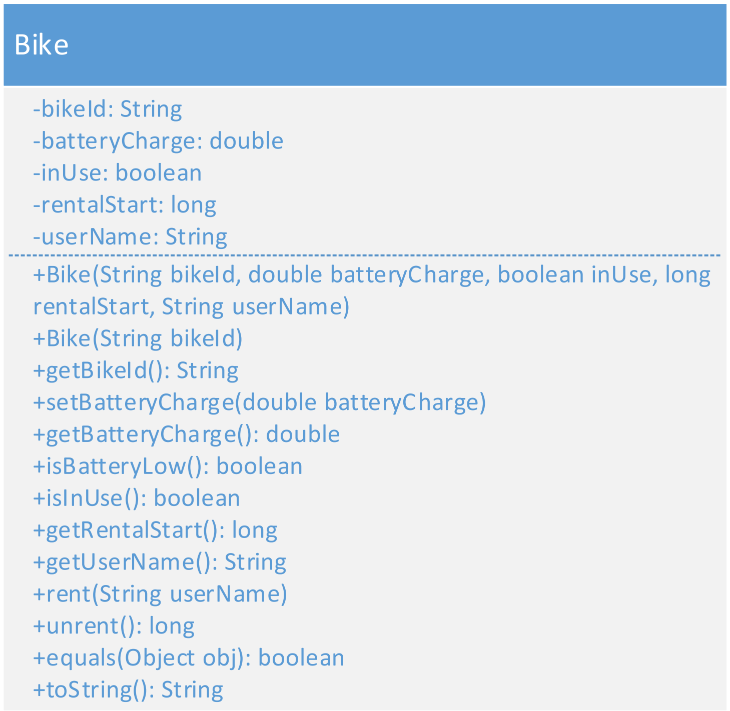
Here are some additional requirements that should be implemented:

- Bikes with very low batteries will be unavailable to rent until such time that their batteries are charged up to more than 20%
- Each battery charge applied (i.e. unit of charge) adds 0.25 to the battery's charge level, which cannot exceed the maximum of 1.0
- Battery drain occurs with usage over time. For this app, loss of charge will be simulated at a rate of 0.002 per rental minute and will be calculated/applied upon return of the bike.
- The cost of each rental will be calculated as $1.00 base fee + $0.25 per rental minute

Material Provided

The accompanying *Bike.java* file contains source code for the Bike class. Add this class to your BlueJ project but please DO NOT alter its source code.

Here is a UML diagram of the Bike class:



```
Bike
-bikeId: String
-batteryCharge: double
-inUse: boolean
-rentalStart: long
-userName: String
------------------------------------------------------------
+Bike(String bikeId, double batteryCharge, boolean inUse, long
rentalStart, String userName)
+Bike(String bikeId)
+getBikeId(): String
+setBatteryCharge(double batteryCharge)
+getBatteryCharge(): double
+isBatteryLow(): boolean
+isInUse(): boolean
+getRentalStart(): long
+getUserName(): String
+rent(String userName)
+unrent(): long
+equals(Object obj): boolean
+toString(): String
```

The accompanying *BikeTest.java* file contains a small demonstration program that uses the Bike class. This program is provided to give you some examples of how to interact with the Bike class as you will need to do similar interaction in your application. Feel free to add the BikeTest class to your project, examine the code, run it, and even modify it if you want to try something out.

Test Plan

Below are **2 separate sample runs** which make up the test plan. Your timing-dependent values will of course be different from these sample runs, which is to be expected. However, please allow for a pause of 5 - 10 minutes between run #1 and run #2 so that your results capture a similar rental duration. User input appears as **red bold underline**. Important output elements are shaded yellow simply for empahasis/convenience:

Sample Run #1

```
*** Welcome to R&R E-Bike Rentals ***

Enter bike data filename: c:/temp/bikes.csv

Bike ID  Battery  Rental Status
-------  -------  ----------------------------------------
   BE11     100%
   BE12     100%
   BE13     100%
   BE14      90%
   BE15     100%
   BE16     100%
   BE17      16%
   BE18     100%
   BE19     100%
   BE20      55%
   BE21     100%
   BE22     100%

What would you like to do (1=Rent, 2=Return, 3=Charge Battery, 4=Exit)?
Enter your option number: 1

Rent a bike. Enter bike ID: BE27
Sorry. Bike BE27 is not found.
Press [Enter] to continue...

Bike ID  Battery  Rental Status
-------  -------  ----------------------------------------
   BE11     100%
   BE12     100%
   BE13     100%
   BE14      90%
   BE15     100%
   BE16     100%
   BE17      16%
   BE18     100%
   BE19     100%
   BE20      55%
   BE21     100%
   BE22     100%

What would you like to do (1=Rent, 2=Return, 3=Charge Battery, 4=Exit)?
Enter your option number: 1

Rent a bike. Enter bike ID: BE17
Bike BE17 has a low battery and cannot be rented.
Press [Enter] to continue...

Bike ID  Battery  Rental Status
-------  -------  ----------------------------------------
   BE11     100%
   BE12     100%
   BE13     100%
   BE14      90%
   BE15     100%
   BE16     100%
   BE17      16%
   BE18     100%
   BE19     100%
   BE20      55%
   BE21     100%
   BE22     100%

What would you like to do (1=Rent, 2=Return, 3=Charge Battery, 4=Exit)?
Enter your option number: 1

Rent a bike. Enter bike ID: be15
Enter customer name or email: steve84@hotmail.com
Bike BE15 rented to steve84@hotmail.com.
Press [Enter] to continue...

Bike ID  Battery  Rental Status
-------  -------  ----------------------------------------
   BE11     100%
   BE12     100%
   BE13     100%
   BE14      90%
   BE15     100%  Rented at 12:43 to steve84@hotmail.com
   BE16     100%
   BE17      16%
   BE18     100%
   BE19     100%
   BE20      55%
   BE21     100%
   BE22     100%

What would you like to do (1=Rent, 2=Return, 3=Charge Battery, 4=Exit)?
Enter your option number: 1

Rent a bike. Enter bike ID: BE15
Bike BE15 is in use and cannot be rented.
Press [Enter] to continue...

Bike ID  Battery  Rental Status
-------  -------  ----------------------------------------
   BE11     100%
   BE12     100%
   BE13     100%
   BE14      90%
   BE15     100%  Rented at 12:43 to steve84@hotmail.com
   BE16     100%
   BE17      16%
   BE18     100%
   BE19     100%
   BE20      55%
   BE21     100%
   BE22     100%

What would you like to do (1=Rent, 2=Return, 3=Charge Battery, 4=Exit)?
Enter your option number: 4

Good bye!
```

Sample Run #2

```
*** Welcome to R&R E-Bike Rentals ***

Enter bike data filename: c:/temp/bikes.csv

Bike ID  Battery  Rental Status
-------  -------  ----------------------------------------
   BE11     100%
   BE12     100%
   BE13     100%
   BE14      90%
   BE15     100%  Rented at 12:43 to steve84@hotmail.com
   BE16     100%
   BE17      16%
   BE18     100%
   BE19     100%
   BE20      55%
   BE21     100%
   BE22     100%
```

CMPP269 Assignment #3 – R&R E-Bikes

```
What would you like to do (1=Rent, 2=Return, 3=Charge Battery, 4=Exit)?
Enter your option number: 3

Charge a battery. Enter bike ID: Ol'Bess
Sorry. Bike Ol'Bess is not found.
Press [Enter] to continue...

Bike ID  Battery  Rental Status
-------  -------  ----------------------------------------
   BE11     100%
   BE12     100%
   BE13     100%
   BE14      90%
   BE15     100%  Rented at 12:43 to steve84@hotmail.com
   BE16     100%
   BE17      16%
   BE18     100%
   BE19     100%
   BE20      55%
   BE21     100%
   BE22     100%

What would you like to do (1=Rent, 2=Return, 3=Charge Battery, 4=Exit)?
Enter your option number: 3

Charge a battery. Enter bike ID: be15
Bike BE15 is rented and cannot be charged.
Press [Enter] to continue...

Bike ID  Battery  Rental Status
-------  -------  ----------------------------------------
   BE11     100%
   BE12     100%
   BE13     100%
   BE14      90%
   BE15     100%  Rented at 12:43 to steve84@hotmail.com
   BE16     100%
   BE17      16%
   BE18     100%
   BE19     100%
   BE20      55%
   BE21     100%
   BE22     100%

What would you like to do (1=Rent, 2=Return, 3=Charge Battery, 4=Exit)?
Enter your option number: 3

Charge a battery. Enter bike ID: be20
1 charge unit applied to bike BE20.
Press [Enter] to continue...

Bike ID  Battery  Rental Status
-------  -------  ----------------------------------------
   BE11     100%
   BE12     100%
   BE13     100%
   BE14      90%
   BE15     100%  Rented at 12:43 to steve84@hotmail.com
   BE16     100%
   BE17      16%
   BE18     100%
   BE19     100%
   BE20      80%
   BE21     100%
   BE22     100%

What would you like to do (1=Rent, 2=Return, 3=Charge Battery, 4=Exit)?
Enter your option number: 3

Charge a battery. Enter bike ID: BE20
1 charge unit applied to bike BE20.
Press [Enter] to continue...

Bike ID  Battery  Rental Status
-------  -------  ----------------------------------------
```

```
   BE11     100%
   BE12     100%
   BE13     100%
   BE14      90%
   BE15     100%  Rented at 12:43 to steve84@hotmail.com
   BE16     100%
   BE17      16%
   BE18     100%
   BE19     100%
   BE20     100%
   BE21     100%
   BE22     100%

What would you like to do (1=Rent, 2=Return, 3=Charge Battery, 4=Exit)?
Enter your option number: 2

Return a bike. Enter bike ID: be14
Bike BE14 does not need to be returned.
Press [Enter] to continue...

Bike ID  Battery  Rental Status
-------  -------  ----------------------------------------
   BE11     100%
   BE12     100%
   BE13     100%
   BE14      90%
   BE15     100%  Rented at 12:43 to steve84@hotmail.com
   BE16     100%
   BE17      16%
   BE18     100%
   BE19     100%
   BE20     100%
   BE21     100%
   BE22     100%

What would you like to do (1=Rent, 2=Return, 3=Charge Battery, 4=Exit)?
Enter your option number: 2

Return a bike. Enter bike ID: BE15
Bike BE15 returned. Minutes used: 9  Cost: $3.25
Press [Enter] to continue...

Bike ID  Battery  Rental Status
-------  -------  ----------------------------------------
   BE11     100%
   BE12     100%
   BE13     100%
   BE14      90%
   BE15      98%
   BE16     100%
   BE17      16%
   BE18     100%
   BE19     100%
   BE20     100%
   BE21     100%
   BE22     100%

What would you like to do (1=Rent, 2=Return, 3=Charge Battery, 4=Exit)?
Enter your option number: 4

Good bye!
```

Development Strategy

Your rental application should be developed as a new Java class with a main method that initiates and controls the program flow. The main method should call on other supporting methods, which you of course need to develop, to implement the desired functionality. Here are the suggested supporting methods to build into your application:

1. **loadBikes**
   - Purpose: Opens and reads the specified data file (e.g. “c:/temp/bikes.csv”) and populates the referenced array/ArrayList of Bike objects, with each successive Bike object getting its attribute values from the corresponding line in the data file.
   - Parameters:
     - Filename for the bike data file
     - Reference to an array or ArrayList that will contain all the Bike objects
   - Notes/tips: Use the File and Scanner classes in this method. You can read individual values from a CSV file using token-based Scanner methods (e.g. next, nextDouble) by overriding the default delimiter. Just call the useDelimiter method once on your file Scanner to set this up. For example:

     ```
     Scanner inData = new Scanner(new File("c:/temp/mydata.csv"));
     // Set token delimiter to either a comma or a newline
     inData.useDelimiter(",|\r\n");
     ```

2. **saveBikes**
   - Purpose: Iterates through the array or ArrayList and writes each Bike object’s field values to the specified data file. This operation overwrites the previous file contents!
   - Parameters:
     - Filename for the bike data file
     - Reference to an array or ArrayList that contains all the Bike objects
   - Notes/tips: Use the PrintWriter class in this method.
3. **showMenu**
   - Purpose: Refreshes the screen (prints the list of bikes followed by menu options), inputs the user’s menu selection, and returns it to the caller.
   - Parameters:
     - Reference to an array or ArrayList that contains all the Bike objects
     - Reference to a Scanner object attached to the keyboard
   - Returns: the user’s selected menu option
4. **findBike**
   - Purpose: Searches the referenced array or ArrayList of Bike objects for a bike having the specified bike ID.
   - Parameters:
     - Reference to an array or ArrayList that contains all the Bike objects
     - The bike ID (String) to be searched for
   - Returns: reference to the matched Bike object, null if no match was found

5. **rentBike**
   - Purpose: Performs the input/processing/output steps specific to menu option 1 - Rent a bike.
   - Parameters:
     - Reference to an array or ArrayList that contains all the Bike objects
     - Reference to a Scanner object attached to the keyboard
6. **returnBike**
   - Purpose: Performs the input/processing/output steps specific to menu option 2 - Return a bike.
   - Parameters:
     - Reference to an array or ArrayList that contains all the Bike objects
     - Reference to a Scanner object attached to the keyboard
7. **chargeBike**
   - Purpose: Performs the input/processing/output steps specific to menu option 3 - Charge battery.
   - Parameters:
     - Reference to an array or ArrayList that contains all the Bike objects
     - Reference to a Scanner object attached to the keyboard

Work with your team to first define the headers for all of your methods, then determine how you will divide the programming tasks among the team members and how you will assemble and test the complete application.

Submission and Grading

Please follow the specific submission requirements from your instructor. For a group assignment this would generally mean that one solution, consisting of your group's final Java source code and accompanying text file with output from the test runs, be submitted to Brightspace.

Submissions will be evaluated according to the following standard:

1. Working Java code (60%)
   - Follows the high level design provided
2. Style (20%)
   - Indentation – consistent
   - Readability – good variable names
   - Documentation
     - Comments at the top which include: Name, date, and program description including details on inputs, processing and outputs (4-5 sentences minimum).
     - Block comments indicating major code sections and what they do. Including a comment block just before each method is good practice!
3. Sample run(s) that matches the provided test plan (20%)
   - Output formatted according to specification (test plan)