**CO518 Assignment 1 Due Date: 25/10/2019 Total Marks: 50**

Lithium is the lightest metal and has the greatest electrochemical potential to provide high energy densities. This means that it is a very attractive metal for high quality batteries in electric cars. Due to climate change lithium has been used in electric vehicles, replacing the nickel-based batteries – that have a memory effect and lose capacity. Lithium-ion batteries are superior to other batteries and have no memory effect. Today, it is the fastest growing and most promising battery chemistry for students to study at universities.

The use of lithium has not come without controversy as mining for “green” electric cars is leaving a stain on the planet when lithium is used. The supply of lithium is limited to high-altitude lakes and bright white salt flats – known as the lithium triangle – Bolivia, Chile and Argentina. The lithium found here is mined in a grid of brine pools. Other sources of lithium can be found in Australia, parts of China, America and Zimbabwe, where it is often found in petalite or traces of it found in igneous rocks and mineral springs. Given the massive climate changes experienced across the world, the jury is still out as to whether lithium provides a sustainable source of energy.

However, you have been tasked with producing an application that acts as a simulator to evaluate lithium samples. This is because lithium can be sourced from rock pools, sea flats and deserts and the quality of the lithium can vary greatly. Your application will be responsible for grading, sorting, and pricing thus providing valuable information to customers that would like to purchase lithium.

Table 1 shows a UML diagram of the simulator application that you are required to develop.

.

Table 1: UML Diagram for Lithium application

Let’s take a look at each class and describe the actions that need to take place to produce a working application.

**GenerateLithium (5 marks)**

* This class is responsible for generating a tray of lithium. The code for the beginning of the class is given to you below so that you can copy it over to your project. You need to complete the rest of the code for this class.

This class must generate a tray of lithium using a 2-dimensional array. For example, it can look as follows:

42 || 2 || 48 ||

20 || 32 || 37 ||

32 || 47 || 32 ||

35 || 16 || 48 ||

27 || 1 || 16 ||

* As seen in the code lithium grades range from 1 to 50.

public class GenerateLithium

{//you may alter this code accordingly as there may be some anomalies

int tray [][] = new int [5][3];

private int grading = 0;

private Random randomGenerator;

/\*\*

\* Constructor for objects of class GenerateLithium

\*/

public GenerateLithium()

{

randomGenerator = new Random();

}

public void generateSample()

{

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 3; j++)

{

grading = randomGenerator.nextInt(50);

//Student to continue coding from here

* The only remaining code that forms part of this class is a method called *printTray()* that prints the generated lithium from *generateSample()* method.

**LithiumGrading (20 marks)**

* This class makes use of the tray of lithium generated in the GenerateLithium class so that the lithium can be graded, as it is important for customers to buy highly graded lithium.
* The class must consist of 2 ArrayLists called lowGrade and highGrade as a method will determine the grade of the lithium.
* There must be a method called *generateGrades()* that accepts the lithium tray as a parameter and must look at each lithium grade and determine whether it is of a high quality or not. Any lithium that is above a value of 25 is considered to be a high grade. The aim of this method is to generate a list that contains all the low graded lithium and another list that contains the high graded lithium.
* There needs to be an additional method that makes use of the bubble sort to sort the low-graded as well as the high-graded lithium in ascending order. This method is called *sortingLithium()*.
* An example of what the output can look like is as follows:

High grade

32

33

38

40

Low grade

2

4

11

12

13

15

16

25

**LithiumPricing (15 marks)**

* Make use of Hashmaps to set the pricing of the lithium as this class is responsible for ensuring that each lithium sample is given a price. The following table illustrates the pricing criteria.

**Grade Price - £**

1 to 9 300.00

10 to 19 600.00

20 to 29 900.00

30 -> 1250.00

* Make use of a method called *setPrice()* to perform the calculations associated with ensuring that each sample of lithium in the tray is given a price accordingly. For example, if a sample in the tray is given a grade of 15 then the price for that sample would be £600.00.
* Include a method called *printPrice()* that prints all the prices of the lithium. The printed price list should show all the lithium found in the tray together with their prices.

For example, a list may look as follows:

32 1250.00

7 300.00

41 1250.00

11 600.00

**BuyLithium (10 marks)**

* This class consists of 2 methods, the first one being *findBestPrice()*. This offers a buyer the chance to enter a price and the method should show all the lithium that is on offer at the buyer’s price or any price lower than what the buyer has stated.
* Make use of a parameter and not the scanner class to input the buyer’s price.
* The output of the method is to show all the lithium grades with their prices as well as indicate to the buyer the number of lithium grades available. For example, a list may include the following if a buyer wants to see all the lithium for sale that is 900.00 or less:

20 900.00

11 600.00

16 600.00

There are 3 choices available to you.

* The second method must be called *findHighQuality()* and should be coded in a similar manner as the *findBestPrice()* method, except that the buyer should enter a value for quality (ranging from 1 to 50) they want to purchase. Again, if the buyer includes the grade of 20, all lithium with a grade of 20 or more should be listed, together with their price. There should be a message that indicates how many sources / grades of lithium were located.

**Hand-in Instructions:**

* The deadline is Monday, 25th October 2019 at 23:55. Submission is via the CO518 Moodle page.
* No late submissions will be accepted.
* Submit a single .zip file containing all the source code, using your login as the file name.

**Plagiarism**

* The work you submit must be your own. We will run checks on all submitted work in an effort to identify possible plagiarism, and take disciplinary action against anyone found to have committed plagiarism.
* Some guidelines on avoiding plagiarism
* One of the most common reasons for programming plagiarism is leaving work until the last minute. Avoid this by making sure that you know what you have to do (that is not necessarily the same as how to do it) as soon as an assessment is set. Then decide what you will need to do in order to complete the assignment. This will typically involve doing some background reading and programming practice. If in doubt about what is required, ask a member of the course team.
* Another common reason is working too closely with one or more other students on the course. Do not program together with someone else, by which I mean do not work together at a single PC, or side by side, typing in more or less the same code. By all means discuss parts of an assignment, but do not thereby end up submitting the same code.
* It is not acceptable to submit code that differs only in the comments and variable names, for instance. It is very easy for us to detect when this has been done and we will check for it.
* Never let someone else have a copy of your code, no matter how desperate they are. Always advise someone in this position to seek help from their class supervisor or lecturer. Otherwise they will never properly learn for themselves.
* It is not acceptable to post assignments on sites such as RentACoder and we treat such actions as evidence of attempted plagiarism, regardless of whether or not work is payed for.
* Further advice on plagiarism and collaboration is also available
* You are reminded of the rules about plagiarism that can be found in the Stage I Handbook. These rules apply to programming assignments. We reserve the right to apply checks to pro- grams submitted for assignment in order to guard against plagiarism and to use programs submitted to test and refine our plagiarism detection methods both during the course and in the future.