CMPSCI 111L Intro to Algorithms and Programming: Java Lab

Programming Project #2 – User Input and Selections (30 points)

Due 3/16/20

CostOfDriving (5 points)

Create a NetBeans project named **Project2**. In Project2, create a main class with the name **CostOfDriving**. In the **CostOfDriving** `main()` method body, write a program that prompts the user to enter the distance to drive, the fuel efficiency of the car in miles per gallon, and the price per gallon then displays the cost of the trip. Note: You will want to use the rounding methods in the Math class (discussed on pages 121 and 122 of Chapter 3) to round the dollar amount to two decimal places.

From the Project Navigator you can run the new program by right clicking on the file name and selecting 'Run File' or if the file is active in the NetBeans code editor window, press 'Shift - F6'. Here is a sample run in NetBeans. Your program's user prompt and final output **must** look similar to this:

```
Enter the driving distance: 900.5  Enter
Enter miles per gallon: 25.5  Enter
Enter price per gallon: 3.55  Enter
The cost of driving is $125.36
```

RunwayLength (5 points)

DO NOT create another project, we are going to add a new file to Project2. From the NetBeans 'File' menu select '**New File**'. In the dialog box under '**File Type**' select '**Java Main Class**' and click '**Next**'. Create a main class with the name **RunwayLength**, verify it is part of Project2 and click the '**Finish**' button. Given an airplane's acceleration a and take-off speed v, you can compute the minimum runway length needed for that airplane to take off using the following formula:

$$length = \frac{v^2}{2a}$$

In the **RunwayLength** `main()` method body, write a program that prompts the user to enter v in meters/second (m/s) and the acceleration a in meters/second squared (m/s^2), then computes and displays the minimum runway length. Here is a sample run in NetBeans. Your program's user prompts and final output **must** look similar to this:

```
Enter speed (meters/second): 60 [Enter]
Enter rate of acceleration (meters/second^2): 3.5 [Enter]
The minimum runway length is 514.286 meters
```

Magic8Ball (20 points)

DO NOT create another project, we are going to add a new file to Project2. From the NetBeans 'File' menu select 'New File'. On the dialog box under 'File Type' select 'Java Main Class' and click 'Next'. Create a main class with the name '**Magic8Ball**', verify it is part of Project2 and click the 'Finish' button. In the **Magic8Ball** `main()` method body, write a program that simulates the popular children's decision-making toy called the Magic 8 Ball.

To do this, the program will first have to prompt the user to enter his or her question. The user's response should be "stored" in a String-type variable named **question**. Use the **`nextLine()`** method on your Scanner-type object to obtain the *entire* line of text entered by the user.

Once the program has read the entire line of text entered by the user, the program should generate a random number between 0 and 19, which represents the 20 responses which are possible for a Magic 8 Ball:

- It is certain
- It is decidedly so
- Without a doubt
- Yes definitely
- You may rely on it
- As I see it, yes
- Most likely
- Outlook good
- Yes
- Signs point to yes

- Reply hazy try again
- Ask again later
- Better not tell you now
- Cannot predict now
- Concentrate and ask again
- Don't count on it
- My reply is no
- My sources say no
- Outlook not so good
- Very doubtful

The program must generate this random number using the **`Math.random()`** method and should "store" the number in an int-type variable named **randomNumber**.

The program should then use a selection (either nested if or switch structure) to "decide" which Magic 8-ball response to display. Here is what your program's user prompt and final output **must** look like. Notice that the user's question must be repeated or "echoed" back after it is input:

```
What is your question?  Will it snow tomorrow? Enter

Your question was: Will it snow tomorrow?

My answer is: Outlook not so good.
```

When you have completed all of the exercises, ZIP the entire project folder and upload this ZIP file to Canvas. In NetBeans, click 'File', select 'Export Project', click 'To ZIP…', select the location where you want the ZIP file to be saved, then click the 'Export' button.

If the IDE you are using does not have an export-project-to-zip option, you may have to manually ZIP the file by navigating to the project folder in a file browser and selecting your OS's "compress folder" option. Under Mac OS, the file browser is called Finder which offers a Compress Folder option. Under Windows, the file browser is called Windows Explorer and it offers a Send to => Compressed (zipped folder) option.