CSE1341 – Lab #2

PRE-LAB [10 points]

Must be done prior to your lab session .

Create a Java program named MathWhiz.java that prompts for five 2-digit whole numbers. It then calculates and displays sum and product of the five numbers. Use *printf* formatting to display a comma at every thousandths digit.

SAMPLE OUTPUT:

```
$ java MathWhiz
Enter five 2-digit whole numbers below

Number 1: 99
Number 2: 99
Number 3: 99
Number 4: 99
Number 5: 99
The sum of the five numbers is 495
The product of the five numbers is 919,965,907
```

Note: highlighted text indicates data entered by user at runtime.

Bring the compiled program (.*class file)* along with the .*java* source file to your lab session. Demonstrate the working program to your lab instructor for pre-lab credit.

LAB [90 points] Due: Saturday 2/23/2019 6:00am

The following programs are all based on the spreadsheet assignments you completed for lab 1. Complete the code for each of them so they compile and run successfully.

1) **[30 points] Gradebook**
   Create the Gradebook you developed in Lab 1 as a Java program. You will prompt the user to enter the 17 grades, then calculate and display the quiz, lab and exam averages, as well as the final grade based on the weightings described in lab 1. Note: To reduce complexity, we will NOT drop the lowest lab and quiz grades. Your output should match the output shown below. Watch your variable types (you will need double for some of these value.) Also, use *printf* to properly format the output of floating point numbers to round to the nearest 1/100th.

Sample run:

```
$ java Gradebook
Welcome to the Gradebook. Please enter grades (0-100) when prompted.

Quiz 1: 80
Quiz 2: 90
Quiz 3: 70
Quiz 4: 100
Quiz 5: 100
Lab 1: 90
Lab 2: 80
Lab 3: 100
Lab 4: 85
Lab 5: 95
Lab 6: 78
Lab 7: 90
Lab 8: 95
Exam 1: 85
Exam 2: 90
Exam 3: 95
Participation: 100

SUMMARY
Quizzes: 88.00 Labs: 89.00 Exams: 90.00 Participation: 100
FINAL GRADE: 89.95
```

Note: highlighted text indicates data entered by user at runtime.

2) **[30 points] Currency Calculator**

Create a variation of the currency converter from Lab 1 as a Java program. Take care to use the correct primitive types for your numeric variables. You will display the conversion table, then prompt the USD amount to convert to the other five currencies and display the results. Use *printf* formatting to align the data.

Sample run:

```
$ java Currency
CURRENCY RATES
Country    Currency         Rate
USA        USD              1.00
UK         GBP              0.77
Europe     EU               0.88
Japan      Yen            109.59
Mexico     Peso            19.13
Vietnam    Dong         23,195.50

Enter amount in USD: $ 45678.12

CURRENCY               AMOUNT
     GBP            35,172.15
      EU            40,196.75
     Yen         5,005,865.17
    Peso           873,822.46
    Dong     1,059,526,832.46
```

Note: highlighted text indicates data entered by user at runtime.

3) [30 points] Commute Calculator

Create a Java version of the commute calculator you created in Lab 1. Prompt the user to enter distance (in miles) target speed (in mph) project rainfall (in inches) and projected number of accidents. All four values should be captured as whole numbers. Calculate the actual average speed by reducinig the target speed by 15% for each inch of rainfall and 10% for each projected accident. Display the actual average speed and duration of the journey. Watch your choice of data types, and use *printf* formatting to round the calculated data to the nearest tenth.

```
$ java Commute
COMMUTE CALCULATOR (Enter all values as whole numbers)

Enter commute distance (miles): 30
Enter target average speed (mph): 65
Enter projected rainfall (inches): 1
Enter projected number of accidents: 2

Actual average speed:  44.2 mph
Commute duration:  40.7 minutes
```

Note: highlighted text indicates data entered by user at runtime.

Grading Rubric

NOTES: Use the given notes as a guide for the program logic. These comments must be included in the programs to explain the logic followed. Each program should compile without errors and should run to produce outputs described for each exercise.

The following points will be discounted if the related element is missing or incorrect:

Reasonable output formatting [20 points]
Proper names for classes and variables [15 points]
Comments [15 points]
Program doesn't compile [20 points]
Source code (java files) missing [15 points]
Executable (class files) missing [15 points]

Plagiarism or collaboration with anyone other than your professor, lab instructor, or CSE help desk personnel will result in no credit for the assignment and possible honor code violation. Plagiarism is inclusion of any line of code that was created by another person, regardless of the source.

Submit the *java* and *class* files via Canvas (as a single zip-file). Include a comment block at the top of each *Java* file that includes your name, student id number, and “Lab 2-Spring 2019”.