COSC 1046 – Fall 2018 - Assignment 2

This assignment is designed to give you some practice and experience writing Java applications using if statements and collecting input from the user. Each question specifies the name that each file/application should have. Please follow this naming convention. When you have completed the assignment compress all of your files into a single archive using Windows (Right Click folder -> Send To -> Compressed Folder) or OS X (Right Click folder -> Compress). Using a 3rd party compression utility such as WinRAR or 7zip may render your files unreadable and un-markable. Submit a single compressed file to D2L. You can resubmit your files as many times as you would like up to the due date and time.

Be sure to include your name and a brief description of your program (as comments) at the top of each file. Pay attention to using good variable names. If you have any questions, please check or post to the forum.

Submit solutions for these questions. Wherever applicable, do your best to reproduce my output exactly. In my sample **bold** indicates user-input.

If you work in partners, submit only one solution and make sure that both partners' names are in ALL files. If only one name appears, only one person will get the grade. No exceptions.

A word in general on assignment grading (for future assignments): If your program produces the displayed output and meets the criteria specified in the question you should expect to receive full marks. Deductions are taken when there are deviations in the output – small deductions for small deviations (calculation error, improper formatting, etc.) while larger deductions are taken for larger deviations (missing output, substantially incorrect values, etc.). Express your creativity in your code, not in your output. Also, deductions will be taken for:

- lack of comments
- poor variable names
- programs that don't run at all (large deduction), so make sure your program runs, even if it is not complete.

If you have any questions or concerns about the grading scheme before you submit an assignment, please post to the forum and ask for clarification. If you have concerns about the grading you've received on an assignment, please email me and I will review the grading form.

Do all of the questions to ensure that you are practicing all concepts but submit solutions to questions 1, and 2 ONLY for grading. It is assumed that you are doing all of the questions and some of them may be referenced in future assignments. Solutions will be provided only for required questions.

Question 1 (GradeCalculator.java)

Write a program that prompts the user to enter three number grades (doubles are probably best here). The program computes the average of the three grades and displays the letter grade for the average. Letter grades are computed as follows (this requires if-statements):

- A+: 90-100
- A: 80-89
- B: 70-79
- C: 60-69
- D: 50-59
- F: < 50

```
C:\Users\aaron\Desktop>java GradeCalculator
Please enter three grades: 60 80 100

Average grade: 80 or A
End of program.

C:\Users\aaron\Desktop>java GradeCalculator
Please enter three grades: 75.5 81.3 19.7

Average grade: 58.83 or D
End of program.
```

Since most grades are between 0 (#sadfaceemoji) and 100 (#highfivegif) we shouldn't proceed with the program if any of the grades are outside of that range. If any of the grades are < 0 or > 100, stop the program with an appropriate message.

```
C:\Users\aaron\Desktop>java GradeCalculator
Please enter three grades: -75.5 81.3 19.7
One of the grades is out of range.
End of program.

C:\Users\aaron\Desktop>java GradeCalculator
Please enter three grades: 75.5 81.3 119.7
One of the grades is out of range.
End of program.

C:\Users\aaron\Desktop>java GradeCalculator
Please enter three grades: 75.5 81.3 19.7

Average grade: 58.83 or D
End of program.
```

Question 2.

(`StringChecker.java`) Write a program that collects three Strings from the user. Determine and display whether the strings are all equal, in increasing or decreasing lexicographical (alphabetical) order or not in any specific order at all.

(Hint: If you are having a hard time getting this to work with Strings try it with integers first. The logic is the same but the syntax is a bit different.)

```
C:\cosc1046\a2\>java StringChecker
Enter 3 strings, one per line:
apples
bananas
coconuts
Your strings are in lexicographical order (ascending) !

C:\cosc1046\a2\>java StringChecker
Enter 3 strings, one per line:
coconuts
bananas
apples
Your strings are in lexicographical order (descending) !

C:\cosc1046\a2\>java StringChecker
Enter 3 strings, one per line:
apples
apples
bananas
Your strings are in lexicographical order (ascending) !

C:\cosc1046\a2\>java StringChecker
Enter 3 strings, one per line:
apples
coconuts
bananas
Those strings are not in any order :( <sad>

C:\cosc1046\a2\>java StringChecker
Enter 3 strings, one per line:
apples
apples
apples
The strings are equal.
```

DO NOT SUBMIT THE FOLLOWING QUESTIONS

Question 3 (SimpleMath.java)

Collect two integer values from the user. Based on these values display the sum, difference, product, average, distance (absolute difference), maximum and minimum. Sample output is provided below:

```
C:\Users\aaron\Desktop>java SimpleMath
Enter two integers:
10
20

Sum:30
Difference:-10
Product:200
Average:15.0
Distance:10
Maximum:20
Minimum:10


C:\Users\aaron\Desktop>java SimpleMath
Enter two integers:
101 52

Sum:153
Difference:49
Product:5252
Average:76.5
Distance:49
Maximum:101
Minimum:52
```

Do the same work outlined above BUT have the output aligned in the following manner:

```
C:\Users\aaron\Desktop>java SimpleMath
Enter two integers:
101 52

Sum:               153
Difference:         49
Product:          5252
Average:            76.5
Distance:           49
Maximum:           101
Minimum:            52
```

Question 4 (BillCalculator.java)

Write a program that takes as input the price of a meal. The program then computes and displays the tax (13%) and the total for the bill. If the meal price is < 0, error and quit the program.

```
C:\Users\aaron\Desktop>java BillCalculator
Enter the price of your meal: $55

Tax: $7.15
Bill with tax: $62.15
```

Prompt the user for a second input, a service-quality integer, 1 for great service, 2 for good service and 3 for poor service. Compute an appropriate tip amount (great = 22%, good = 18%, poor = 10%) based on the before-tax meal cost and add it to the after-tax total. If the tip option is less than 1 or greater than 3 exit the program.

```
C:\Users\aaron\Desktop>java BillCalculator
Enter the price of your meal: $100
How was the service?  1=great, 2=good, 3=poor:1

Tax: $13.0
Bill with tax: $113.0
Tip: $22.0
Total bill (including tip): $135.0


C:\Users\aaron\Desktop>java BillCalculator
Enter the price of your meal: $100
How was the service?  1=great, 2=good, 3=poor:3

Tax: $13.0
Bill with tax: $113.0
Tip: $10.0
Total bill (including tip): $123.0
```

Question 5. (BooleanPractice.java)

Write a java script that determines whether a user-entered **integer** is odd or even and whether it is evenly divisible by 3, 4, both 3 and 4 or neither 3 nor 4 (see my output below for clarification). Assume the user will enter a positive integer. The output from my solution is shown below. Do your best to duplicate my output. Below is my output from running the program 5 times.

Use boolean operators (&& or ||) in your solution.

```
C:\Users\aaron\Desktop>java BooleanPractice
Enter an integer:12
12 is even.
12 is divisible by 3 and 4.

C:\Users\aaron\Desktop>java BooleanPractice
Enter an integer:9
9 is odd.
9 is divisible by 3.

C:\Users\aaron\Desktop>java BooleanPractice
Enter an integer:16
16 is even.
16 is divisible by 4.

C:\Users\aaron\Desktop>java BooleanPractice
Enter an integer:13
13 is odd.
13 is neither divisible by 3 nor 4.

C:\Users\aaron\Desktop>java BooleanPractice
Enter an integer:132
132 is even.
132 is divisible by 3 and 4.
```

Question 6. (**`BooleansAndStrings.java`**)

Write a program that collects three Strings from the user. Display the three strings in alphabetical order regardless of the order in which they were input.

Use boolean operators && and/or || in your solution.

(Hint: If you are having a hard time getting this to work with Strings try it with integers first. The logic is the same but the syntax is a bit different.)

Question 7. (NumberConverter.java)

Write a program that reads a five-digit integer, such as 12345, and then displays it, one digit per line as shown below. There are two ways to do this – one is numerically and another using String methods. Your application should do it both ways.

```
C:\Users\aaron\Desktop>java NumberConverter
Enter a five-digit integer: 12345
Mathematical Solution:
1
2
3
4
5
String method solution:
1
2
3
4
5
```

Your prompt tells the user to enter a five-digit integer. If they enter in an integer that is not 5-digits in length, have the program end with an appropriate error message. (You'll need to wrap what you did in the first part in an if-statement here.)

```
C:\Users\aaron\Desktop>java NumberConverter
Enter a five-digit integer: 10
Sorry, incorrect number of digits.

C:\Users\aaron\Desktop>java NumberConverter
Enter a five-digit integer: 100000
Sorry, incorrect number of digits.

C:\Users\aaron\Desktop>java NumberConverter
Enter a five-digit integer: 31337
Mathematical Solution:
3
1
3
3
7
String method solution:
3
1
3
3
7
```