Programming Assignment: Midterm Project

A Simple Calculator

5185 Programming with Java for Beginners
Instructor: Bineet Sharma

Summary: Write a console program (character based) to do simple calculations (addition, subtraction, multiplication and division) of two numbers, using your understanding of Java.

Description: You need to write a program that will display a menu when it is run. The menu gives five choices of operation: addition, subtraction, multiplication, division, and a last choice to exit the program. It then prompts the user to make a choice of the calculation they want to do. Once the user selects the operation, it will check for valid menu choices (and give an appropriate message if a wrong choice was selected) and then prompts the user to enter two numbers, separated by a space. If the user enters valid numbers, it will do the operation desired, and then displays the result. If the user enters invalid numbers, it displays an error message and asks for the correct input. After displaying the result, it displays *Press enter key to continue.* Once enter key is pressed, it displays the menu again. The program repeats until the user selects the choice to exit.

Sample Run of the Program:

Sample Run of the program:

```
Welcome to <John Doe’s> Handy Calculator

        1.  Addition
        2.  Subtraction
        3.  Multiplication
        4.  Division
        5.  Exit

What would you like to do?  3

Please enter two floats to multiply, separated by a space: 24.0  4.0
Result of multiplying 24.00 and 4.00 is 96.00.

Press enter key to continue ….
```

```
Welcome to <John Doe’>s Handy Calculator

            1.  Addition
            2.  Subtraction
            3.  Multiplication
            4.  Division
            5.  Exit

What would you like to do?  4

Please enter two floats to divide separated by a space: 2.0   0
You can’t divide by zero please re-enter both floats: asasfs asdfasfas
You have entered invalid floats please re-enter: 16.0  4.0
Result of dividing 16.0 by 4.00 is 4.00.

Press enter key to continue ….

Welcome to <John Doe’>s Handy Calculator

            1.  Addition
            2.  Subtraction
            3.  Multiplication
            4.  Division
            5.  Exit

What would you like to do?  5

Thank you for using <John Doe’>s Handy Calculator
```

Note:

1) Replace the <*John Doe*’s> with your name
2) Make sure to properly word the output for the choices and result. If user selects ‘1’ from the menu, the prompt should be for ‘Addition’ not multiplication as shown in the example.
3) *3, 24.0 4.0* are shown in the example to emphasize that it is entered by the user and not part of the program. Underline and italics are not a requirement.
4) Your program should allow input of integer and decimal numbers (floats). The output should always be in decimals, with two decimal digits as precision.
5) Make sure your program will continue displaying the menu after the result is shown and user has pressed the enter key. Your program will exit only when user selects 5.
6) If the user selects an option other than 1-5, show a message that they must select a number between 1 and 5. Give them a chance to re-enter. Continue until a valid number is entered.

7) If they enter invalid values instead of numbers (e.g. strings), provide an error message and give them chances to re-enter. Continue until valid numbers are entered. You don't need to worry about range of floats.
8) Make sure to catch the divide by zero issue. In the case of a division choice, the second number should not be a zero. The users should be given a second chance to enter other than a zero.
9) You should decompose your program in many static methods. So, you can call these static methods from main method to do the functionality required. Your class need not to be instantiated with a 'new'. All methods are directly called from main of same class because they are static methods.

Score: Maximum score is 100 and it is divided into the following categories:

1) Program compiles, runs, and provides the correct answers as described above. (70%)
2) The design is modular and has a number of static methods doing part of the work. It should have proper data structures and use descriptive member variable names. (20%)
3) Proper error trappings & exception handling is implemented, and is adequately commented and formatted. (10%)

Submission requirement: Submit your source code. Submit all java file (Calculator.java only) through UCSC web-portal (as an attachment) before the deadline. You DO NOT need to submit anything else.

Naming requirements:

Main Class: The public class which does the calculation must be named as **Calculator** and stored in Calculator.java file**.**