CS 1803

Pair Homework 3 – Calculator Pair Fun

Due: Friday, February 11th, before 6 PM
Out of 100 points

Files to submit: **1. HW3.py**

<u>This is a PAIR PROGRAMMING Assignment: Work with your partner!</u>
For pair programming assignments, you and your partner should turn in identical assignments. List both partners names at the top. Your submission must not be substantially similar to another pairs' submission. Collaboration at a reasonable level will not result in substantially similar code. Students may only collaborate with fellow students currently taking CS 1803, the TA's and the lecturer. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for others.

For Help:
- TA Helpdesk – Schedule posted on class website.
- Email TA's or use T-Square Forums

Notes:
- **Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).**
- **Do not wait until the last minute** to do this assignment in case you run into problems.

User Interaction

You will write a few python functions for practice with user interaction and string and list processing. In your HW3.py file, include a comment at the top with your names, section, GTID/Emails, and your collaboration statement. Also include each of the following functions.

1. getNumber
2. getMathOp
3. getYesNo()
4. doCalculation()
5. calculator()

Function Name: **getNumber**

Parameters:

string – A String which will be shown to the user to "prompt" them.

Return Value:

A float.

Test Cases:

getNumber("Please enter a Number:") →

Please enter a Number: 10

If the user typed a valid number (as above), return the float the user entered; in this case, getNumber returns 10.0

getNumber("Please enter a Number:") →

Please enter an Number: Hello World
Not a valid number, try again…
Please enter a Number: +-/*
Not a valid number, try again…
Please enter an Number: 5
getNumber returns 5.0

Description:

Write a function that will accept one parameter, a string, and show it to the user. The function should then get input from the user and convert it into a float. Return the float value. If the user does not enter a proper number, your function must tell them that they entered an invalid number using the warning: "Not a valid number, try again..." Then, show the prompt again, and allow them to try and type another number. Keep repeating until the user enters digits that can be successfully converted to a float.

Function Name: **getMathOp**

Parameters:

string - a String which will be shown to the user to "prompt" them.

Return Value:

A string with one of four values: "+", "-", "*", or "/"

Test Cases:

getMathOp("Please enter one of the four permitted math operations:") →

Please enter one of the four permitted math operations: +

If the user typed a valid math operation, return the string representing the math operation; in this case, getMathOp returns +

getMathOp("Please enter one of the four permitted math operations:") →

Please enter one of the four permitted math operations: Hello World
You may only enter one of the characters: +-*/
Please enter one of the four permitted math operations: +-/*

You may only enter the one of the characters: +-*/
Please enter one of the four permitted math operations: 17
You may only enter the one of the characters: +-*/
Please enter one of the four permitted math operations: *
getMathOp returns *

Description:
Write a function that takes in a string to use as a prompt and shows it to the user. Then get input from the user. If they entered a single sign out of the set +,-,*,/, return it. If they entered anything else, give them a warning: "You may only enter one of the characters: + - * /" and then repeat until they enter a correct option.

Function Name: **getYesNo**

Parameters:
string – a string to show the user as a prompt.
Return Value:
boolean – True if the user entered Y/yes/Yes/YES/y, etc...
False if the user entered N/NO/No/nO/n, etc...
Test Cases:

```
>>>getYesNo("Do you want to continue?")
Do you want to continue? I don't know
Not a valid answer, enter Y or N.
Do you want to continue? Yes
```

Returns ---> True

Description:
Write a function that takes in a string as a prompt to the user. It should display that string and get input from the user. If the user enters Y, or y, or YES, or Yes, or yeS, or YeS or yEs, or any other permutation of lower and upper case letters that spells out a yes, return true. If they enter N, or n, or NO, or no, return False. If they enter anything else, give them the error "Not a valid answer, enter Y or N." and then repeat the question until they give a valid answer.

Function Name: **doCalculation**

Parameters:
none
Return Value:
Float – Result of calculation.
Test Case:

```
 >>> doCalculation()
Enter your first number: 3
Enter your math operation: +
Enter your second number: 5
```

Returns → 8.0

Description:
Write a function that gets an integer, an operation, and a second integer from the user. After getting those three items (it may call other functions that you have written to do this work) it should perform the appropriate calculation and return the result. Your function must deal properly with invalid inputs, which is most easily done by making use of the previous functions you have already written. Note that your result must be converted to a float, even if it was an integer.

Function Name: **calculator()**

Parameters:

none

Return Value:

none

Test Cases:

```
Enter your first number: 3
Enter your math operation: +
Enter your second number: 5
Your answer is: 8.0
Would you like to perform another calculation? (y or n) Y
Enter your first number: 10
Enter your math operation: *
Enter your second number: 2
Your answer is: 20.0
Would you like to perform another calculation? (y or n) N
The results of your last 2 calculations were: 8.0   20.0
Goodbye!
```

Description:
Write a function that uses your other functions to allow the user to do multiple calculations. Your function must deal properly with invalid inputs, which is most easily done by making use of the previous functions you have already written. After each calculation, ask the user if they want to do another calculation. Be sure to save the results of each calculation as they go!

Once the user says they do not want to do any more calculations, print a final message that states the total number of calculations they performed, and lists the results (final answer) of each calculation before saying “Goodbye!”

Grading:

You will earn points as follows for each function that works correctly according to the specifications.

getNumber		**20**
properly handles negative numbers: "-5"	1	
Properly handles leading spaces: " 35.5"	1	
Properly handles trailing spaces: "34 "	1	
Properly shows prompt	5	
Properly returns answer	5	
Properly handles invalid input: "ten"	7	
getMathOp		**20**
Properly shows prompt	4	
Property handles leading spaces: " +"	3	
Property handles trailing spaces: "+ "	3	
Property handles invalid input: "times"	10	
getYesNo()		**20**
Properly shows prompt.	3	
Properly handles leading spaces: " Yes"	3	
Properly handles trailing spaces: "Y "	3	
Properly handles invalid input: " Nope"	11	
doCalculation()		**20**
Does the right calculation	10	
Returns the proper value (as float)	10	
calculator()		**20**
Allows multiple calculations	5	
Correctly allows user to quit	3	
Prints correct messages	2	
Remembers /prints number & value of results	10	