CMPSC 102
Discrete Structures
Fall 2018

Lab 2 Assignment:
Building your Own Calculator Using Python
Submit deliverables through your assignment GitHib repository bearing your name. Place source code in `src/` and `output` in output/ directories

Objectives

To enhance the understanding of Python by completing a program designed with functions. To learn how to make function calls, performs conditional checks using `if` statements. To practice accepting user-entered data and returning calculated values.

GitHub Starter Link

https://classroom.github.com/a/ysWJFmb_

To use this link, please follow the steps below.

- Click on the link and accept the assignment.
- Once the importing task has completed, click on the created assignment link which will take you to your newly created GitHub repository for this lab.
- Clone this repository (bearing your name) and work on the lab locally.
- As you are working on your lab, you are to commit and push regularly. You can use the following commands to add a single file, you must be in the directory where the file is located (or add the path to the file in the command):
  - `git commit <nameOfFile> -m ‘‘Your notes about commit here’’`
  - `git push`

  Alternatively, you can use the following commands to add multiple files from your repository:
  - `git add -A`
  - `git commit -m ‘‘Your notes about commit here’’`
  - `git push`

Reading Assignment

Please read Chapters 4 (Stavely) and Chapter 1 (Saha) in the course book and the third week's slides.

Handed out: 12^{th} Sept 2018

Additional Materials

Please locate your `src/` directory for this lab where you will find your source code file:

- A partially completed Python source code; `calculator.py`.

About this Lab

In this lab, you will be designing a command-line calculator application in Python which allows a user to enter two numbers and then supply a desired mathematical operator (i.e., "+, -, *, / and %"). There is a conditional check to determine which operator has been entered and then the two numbers are sent to the code that completes the calculation for the operator. The output of this program is shown in Figure 1.

```
------------------------------------------------
|   This is a program to compute the addition,  |
| subtraction, multiplication, division and     |
| modulus of two user-entered numbers.          |
------------------------------------------------
Enter the first number in your equation : 3
  *Your response is : 3
Enter the second number in your equation : 8
  *Your response is : 8
Select an operator (+, -, *, / or %) : *
  *Your response is : *
*Welcome to the doCalc() function.
 The result of << 3 * 8 >> is : 24
```

Figure 1: The output of your program should look exactly like the above for each of the mathematical operators that the user will enter.

You are given partially working code for this assignment that you will find in your src/ directory. Sadly, the part of the code which is required to complete the assignment was eaten by my dog, featured in Figure 2. Namely, he removed one of the functions called `doCalc()` which is called by the program's `main()` function. The `doCalc()` function determines which mathematical operator has been requested by the user and then performs the particular calculation using the inputted numbers. This function returns the calculated value back to the `main()` function. Please openand edit the partial code which is given to you to edit in your `src/` directory. The code is also shown below.

Figure 2: The Beagle who ate the `doCalc()` function in the source code for the command-line calculator application.

Partial Code for Assignment

```
#!/usr/bin/env python3

# Date = 11 Sept 2018
# Version = i
# OriginalAuthor = Oliver Bonham-Carter

# Description: A basic calculator: This program asks the users to enter two numbers.
# Then the user will enter a mathematical operator to be applied to the numbers for
# the calculation.

def getResponse(prmpt, task ="string"):
# Handles the user input aspect for the program
# prmpt: the string of the question to ask the user
# task: if the parameter for this is "float" then return a float, otherwise
# return a string

    response_str = input(prmpt)
    if task == "float":
        return int(response_str)
    else:
        return response_str
#end of getResponse()




    ##########################################
    # doCalc() has been eaten !!!!!          #
    # ToDo: design op-check and calc function #
    ##########################################
```

```
def main(): # driver function
    print("  ------------------------------------------------")
    print("  |   This is a program to compute the addition,  |")
    print("  | subtraction, multiplication, division and     |")
    print("  | modulus of two user-entered numbers.          |")
    print("  ------------------------------------------------")
    prmpt = "  Enter the first number in your equation : "
    num1_flt = getResponse(prmpt,"float") # specify a float to return
    print("    *Your response is :",num1_flt)

    prmpt = "  Enter the second number in your equation : "
    num2_flt = getResponse(prmpt,"float") # specify a float to return
    print("    *Your response is :",num2_flt)

    prmpt = "  Select an operator (+, -, *, / or %) : "
    op_str = getResponse(prmpt) # since no second option is added, we return a string.
    print("    *Your response is :",op_str)

    print("   The result of <<",num1_flt, op_str,num2_flt,">> is :",
doCalc(num1_flt, num2_flt, op_str)) # call function to perform the calculation
    print("")
#end of main()



main() # begin the program
```

Functions

The functions of the code are the following.

- `main()`
  - Main function that drives the program. Calling this function launches the program.
  - No parameters
  - Calls all functions: `getResponse()` (to get the user's information) and `doCalc()` (to perform the calculation and get the result).
- `getResponse()`

  - Allows users to enter numbers and the operator into the program
  - Two parameters: `prompt`; the text for the `input()` function to get numbers or operators from the user, and the `task` flag which determines whether a returned value is a float-type or a string-type.
  - First usage of function's return: `num2_flt = getResponse(prmpt,"float") # specify a float to return and assign to num2_flt`
  - Second usage of function's return: `op_str = getResponse(prmpt) # since no second option is added, we return a string by default to assign to op_str`.

- `doCalc()`
  - Three parameters: `num1_flt` and `num2_flt` (float-type) of the first and second numbers. The `op_str` (string-type) of the mathematical operation to perform on the numbers.
  - Is called by the `main()` function which passes its parameters to it: `num1_flt`, `num2_flt` and `op_str`.
  - Has conditional statements to determine which mathematical operator has been selected by the user.
  - Performs the correct calculation for the operator using float-types.
  - Returns the calculation to the `main()` function for outputting to the screen.

Required Deliverables

Submit deliverables through your assignment GitHib repository bearing your name. Place source code in `src/` and the output text file `output/` directories.

1. Your completed and working python code that you created by editing the file *calculator.py* in the `src/` directory. Please be sure to add your coding comments for your developed function to explain its functionality!
2. A text file (called "results.txt") in the `output` directory, containing the text of the exact output of your program.
3. You should determine that your code works for all inputs and so your output should show that all operators work. Please run tests for each mathematical operator (i.e., "+, -, *, / and %").