1. Write your own calculator program called '**C:\lab\calc.py'** with add, divide, multiply, subtract, and modulus functions.

Test your functions by passing in the following parameters.

add(11, 5)

divide(11,5)

multiply(11, 5)

subtract(11, 5)

modulus(11, 5)

2. You did add Docstrings to your functions? If not, add appropriate Docstrings to each function. To test, start IDLE and load your script (File/Open), then run (<F5>). Then in Python Shell type:

>>> help(add)

>>> help(multiply)

3. Modify the **calc.py** script to allow the add and multiply functions to accept a variable number of parameters. Test the modified functions by passing in the following parameters.

add(11, 5, 33, 15, 6.5)

multiply(11, 5, 33, 15, 6.5)

Stretch

4. In the following exercises, you will use existing demonstrations and code, and improve by creating and using reusable functions.

Open the 'C:\labs\search250.py' script you created in the Regex section. Modify the script so that it has a reusable function called search_movie() which accepts one parameter - the Regex pattern to search. Test the function by calling it from the main() function with the same patterns from the previous lab exercises.

5. Open the 'C:\labs\searchWords.py' script. It searches for regular expressions patterns in the file 'C:\labs\words' and displays them to the Python shell. Modify the script and create a variadic function called search_pattern() which can accept one regex pattern followed by one or more files.

a. If no files are given, then default to 'C:\labs\words'.

b. Iterate through each file printing out lines that match.

Test with the following function calls in the main() function:-

```
search_pattern(r"^([A-Z]).*\1$")
```

```
search_pattern(r"^([A-Z]).*\1$", r"C:\labs\words", r"C:\labs\words")
```