**Description:**You will become familiar with the Spyder IDE and begin writing code using data types and expressions.

* Complete the following assignments and submit by Sunday September 6 by 11:59PM:

* 1. Write a Python multiprocessing program that runs all five functions below. Use a time.counter to tests the amount of time it takes all functions to run synchronously and then concurrently

def add(x, y):

    return x + y

def subtract(x, y):  
    return x-y

def multiply(x, y):  
    return x \* y

def divide(x, y):  
    if y==0:  
        raise ValueError('Cannot divide by zero!')  
    return x / y

def modulus(x, y):  
    return x % y

2. The Luhn algorithm is used to verify whether a credit card number is valid or not. You will need to create a program that uses regular expressions to find the credit card information from a string and then runs the Luhn Algorithm to verify the credit card number. Use the test below.

"Please use my credit card number. It is Visa # 37562198673 with an expiration date of 08/19/2030. The CVS number is 854."

The Luhn algorithm is as follows:

* + - 1. Assume you have a number as: 3 - 7 - 5 - 6 - 2 - 1 - 9 - 8 - 6 - 7 - 3
      2. Now starting from the rightmost digit i.e. check digit, double every second digit. New number will be: 3 - 14 - 5 - 12 - 2 - 2 - 9 - 16 - 6 - 14 - 3
      3. If double of a digit is more then 9, add the digits. So the number will become: 3 - 5 - 5 - 3 - 2 - 2 - 9 - 7 - 6 - 5 - 3
      4. Sum all the numbers together and find the modulus (%). If the modulus is equal to zero, the number is valid

3. In this assignment you are to create a Pickle function that will take pickle the following dictionary:

{'Colorado': 'Rockies', 'Boston': 'Red Sox', 'Minnesota': 'Twins',

'Milwaukee': 'Brewers', 'Seattle': 'Mariners'}

You will then need to run a client/server socket to transfer the file from the client to the server using the following code:

[ClientSocket.py](https://ucsdextension.instructure.com/courses/2191/files/2744083/download?wrap=1)[Preview the document](https://ucsdextension.instructure.com/courses/2191/files/2744083/download?wrap=1)    [SocketServer.py](https://ucsdextension.instructure.com/courses/2191/files/2744082/download?wrap=1)[Preview the document](https://ucsdextension.instructure.com/courses/2191/files/2744082/download?wrap=1)

Once the code is received by the server you will need to unpickle the file in order to recover the dictionary.

Show all of your work and results with screenshots.

\*\* Note: Place both the Client Server, the pickled file on the desktop. I would recommend that you place the Socket Server within a separate file on the desktop so that you don't overwrite the file you are sending.