INSY 5336 001
Python Programming
Fall 2019
Homework 1 (50 points)
Due Date: September 15, 2019, 11:59 pm CST (no exceptions)

The following guidelines should be followed and will be used to grade your homework:

- All code to be implemented and submitted as a jupyter notebook (.ipynb) file. Submit a single ipynb file.
- This is an individual homework assignment, no group submissions will be accepted. If you discuss in groups, please write your code individually and submit.
- Sample runs shown in the question should be used as a guide for implementation. However extensive testing needs to be done on your code to deal with all test cases that might possibly be executed.
- The high level algorithm of how you are solving the problem should be documented in the cell preceding the code in markdown language.
- The instructions for running of each cell and the expected results should be documented in the cell preceding the code using markdown language.
- Every code segment in the jupyter notebook cells should be well documented with comments. Use # in the code to provide comments and they should explain the algorithm step and what the code segment is doing.
- Error checking in your code is very important and differentiates a high quality programmer from a low quality one. Hence you should account for invalid user inputs, infinite loops, out of range results, etc. and resolve them by appropriate error messages. Use try/except blocks, if statements and other python code constructs to deal with unexpected errors and deal with them gracefully. **The homework will be graded for robustness of your code.**

1. (10 points) Implement a program that requests three **strings** from the user. Your program should concatenate the first two strings (string1 + string2) and compare the concatenated string with the third string. If they are equal, your program should print "They are equal", otherwise, the program should concatenate the two strings in the reverse order (string2 + string1) and compare it with the third string and print "They are equal". If this second test also fails, your program should print "They are not equal". A partial sample run is shown below:

```
Enter string 1: bye!
Enter second string: Good
Enter third string: Goodbye!
The two strings are equal

Enter string 1: 12
Enter second string: bye!
Enter third string: bye!12
The two strings are equal

Enter string 1: 12
Enter second string: bye!
Enter third string: Goodbye!
The two strings are not equal
```

2. (10 points) Write a Python program that reads in 3 integers or 4 integers (user first determines 3 or 4 integers) and displays the following:
    a. the average of the numbers (two decimal places suggested)
    b. the maximum of the numbers
    c. the minimum of the numbers
    d. the median of the numbers. Note that if there was 2 numbers that qualify for the median, you will calculate and display the mean of the 2 numbers

    Do NOT use statistical functions such as mean(), median(), max() or min(). You will not receive credit for the question if you do so.
    A sample run is shown below:

```
Enter set of 3 or 4 integers: 3
Enter first number: 7
Enter second number: 10
Enter third number: 1
The average of 7, 10 and 1 is: 6.0
The maximum of the three numbers is:  10
The minimum of the three numbers is: 1
The median of the three numbers is: 7

Enter set of 3 or 4 integers: 4
Enter first number: 7
Enter second number: 10
Enter third number: 1
Enter fourth number: 6
The average of 7, 10 ,1 and 6 is: 6.0
The maximum of the four numbers is:  10
The minimum of the four numbers is: 1
The median of the four numbers is: 6.5
```

3. (10 points) An integer, greater than 1, that is only divisible by 1 and itself is called a prime number. All other numbers greater than 1 are called composite numbers. The integers 0 and 1 are neither prime nor composite. Write a python program that requests a positive integer from the user, determines if it is a prime, composite or neither prime or composite and prints the message. You can choose to use iterative loops to repeatedly run this script or have the user run the script

for every input. Please specify instructions to run the code in markdown language above the code cell.
Sample runs are shown below:

```
Enter positive integer: 3
It is a prime
Enter positive integer: 4
It is a composite
Enter positive integer: 1
It is neither prime nor composite
Enter positive integer: -5
Error: You did not enter a positive integer
Enter positive integer: -2.5
Error: You did not enter a positive integer
Enter positive integer: Hi
Error: You did not enter a positive integer
```

4. (10 points) Implement a Python function SimpleInterest() that takes in 3 arguments:
    a. A principle amount (float)
    b. Interest Rate (0 to 100% as a float)
    c. Year (integer)

   Your function should return the interest amount.
   The formula for a simple interest is (Principle*InterestRate*Years)/100

   Write a main program that requests the principle amount, Interest rate and year from the user. Also in the main program you should call your defined function SimpleInterest() that takes these 3 arguments and display the results.

   Write your function definition in one cell and the function calling main program below it or in a different cell in Jupyter Notebook. I will be testing this program by modifying your program that calls this function.

5. (10 points) Write a program that takes as input two opposite corners of a rectangle: (x1,y1) and (x2,y2) – float or integer only. Finally, the user is prompted for the coordinates of a third point (x,y). The program should print Boolean value True or False based on whether the point (x,y) lies within the rectangle. At the end of each run, the user should be prompted to ask whether then want to continue.
   Note that the rectangle could fall anywhere in the 2X2 plane of real numbers.

   Sample runs are shown below

```
Enter x1: 1
Enter y1: 3
Enter x2: 10
Enter y2: 6
Enter x: 4
Enter y: 4
True
Do you want to continue: Y

Enter x1: 1
Enter y1: 3
Enter x2: 10
Enter y2: 6
Enter x: 4
Enter y: 2
False
Do you want to continue: N
Goodbye!
```