CSE 1321L: Programming and Problem Solving I Lab
Assignment 4 – 100 points
Methods

What students will learn:

1) Designing methods
2) Parameters and returning information
3) Calling methods
4) Problem-solving

Overview: First, we need to talk about terminology. In the world of programming, depending on context, you'll hear the following terms that generally mean the same thing: **method**, **function**, **procedure**, and **subroutine**. Though there are differences, we're going to consider all of these the same thing. In general, methods allow us to group individual lines of code together into one logical unit and reuse it as many times as we want. We have the option of sending pieces of information to methods as a starting point for calculations; those pieces of information are called **parameters**. If a method does calculations and needs to return the results, it can do that by using a **return** statement (after changing the return type to something other than void). Also keep in mind that code execution will no longer start at the top of your code and finish at the bottom. When you want that unit/method to execute, you call it by its name. That method then runs, and when done, execution continues from where it was called..

Assignment 4A: *What's your grade?* In this class (as well as the lecture class), we calculate grades using a process called "weighted averages". The formula works by averaging all the games in a particular category, then multiplying the result by a percentage (the "weight"). All categories are then added up to give you your final grade. In our class, that would look like this:

Final Grade = (average(Labs) * 0.10) + (average(Assignments) * 0.40) + (average(Midterm) * 0.20) + (average(Final) * 0.30)

For this assignment, you will build a program that can calculate your grade in CSE 1321L using the formula above. You will ask users to input the following information:

1) 12 individual lab grades (since the lowest is dropped per FYE policy)
2) 7 assignment grades
3) 1 midterm grade
4) 1 final exam grade

You will then create and use **two** methods to calculate your grade:

- A method to calculate the average value of a category. The method should take in the name of the category and twelve float variables. Based on the category, you should average only those parameters and ignore the rest (e.g. if the category variable is "Assignment", you should average the first 7 parameters and ignore the rest).
  Use default parameters like so:

```
def example(one, grade1=-1, grade2=-1, grade3=-1):
    #Code here

example("Text", 97) #Since the other parameters have defaults, they can
                    #be left off
```

This function returns a float value for the average of the category.

- A method to calculate the weighted points of a category. The method should take in the returned value from the previous method and a float value for the category's percentage (e.g. 0.4) and return a float value for the weighted points of the category.

You will call the same methods for each of the four grade categories – the functions should be flexible enough to work for any values. Once you have all four calculated, add them up and print the results.

Sample Output:

```
[CSE 1321L Grade Calculator]
Labs
Enter Grade #1: 70
Enter Grade #2: 90
Enter Grade #3: 100
Enter Grade #4: 56
Enter Grade #5: 70
Enter Grade #6: 98
Enter Grade #7: 105
Enter Grade #8: 67
Enter Grade #9: 100
Enter Grade #10: 78
Enter Grade #11: 91
Enter Grade #12: 92
Weighted Points: 7.56

Assignments
Enter Grade #1: 90
Enter Grade #2: 100
Enter Grade #3: 78
Enter Grade #4: 90
Enter Grade #5: 95
Enter Grade #6: 67
Enter Grade #7: 100
Weighted Points: 35.429

Midterm
Enter Grade #1: 98
Weighted Points: 19.6

Exam
Enter Grade #1: 100
Weighted Points: 30.0

Your final grade for CSE1321L is: 92.59
```

Assignment 4B: *Caesar Cipher.* One of the simplest methods to encrypt a message is to "shift" the letters by a literal value. For example, if we take the letter 'A' and shift it by 2, we get 'C'. This "substitution cipher" gets its name from the Roman emperor Julius Caesar, who was one of the first person known to encode secret messages this way.

For this assignment, you will create a program that prompts the user for a message to encode, as well as an offset to encrypt it with. You will then create and use the following methods:

1. A validation method that takes in the message (as a string) and the offset integer entered by the user. The method should return "true" if the offset is between 0 and 26 inclusively and the message contains only letters and spaces. If it fails either criteria, it should return "false".

2. If the validation method returns true, a second encryption method should be called. This method also takes in the message (as a string) and the offset integer entered by the user. It should convert the message to UPPERCASE, encrypt every letter in the message, and leave the spaces as they were. It should then return the encrypted message.

Hint: The ord() function will give us the ASCII integer value of a particular letter, and the chr() function will convert a number to a string. How could we use these functions in this assignment?

Hint: What happens if our offset makes the letter go past 'Z'? There are several ways to deal with this – we can either use the modulus operator creatively, or use a loop to add the offset value one at a time. In the latter case, if the letter becomes greater than 'Z' (note the single quotes), we can reset it back to 'A' and keep going.

Sample Output:

```
Enter your message:
KSU Class of 2028!
Enter the offset value: 14

Sorry, we can only process messages with letters and spaces, and the offset
must be between 0 and 26.

Do you want to encrypt another message?: Y
Enter your message:
First Year Experience at KSU
Enter the offset value: 13

Your secret message is...
SVEFG LRNE RKCREVRAPR NG XFH
Do you want to encrypt another message?: n

Closing out...
```

Submission:

1. You will submit 2 separate files
2. File names must be correct.
3. Upload all files (simultaneously) to the assignment submission folder in Gradescope.