**CMPSC-132: Programming and Computation II**
Fall 2019

# Lab #3

Due Date: 09/13/2019, 11:59PM

*Read the instructions carefully before starting the assignment. Make sure your code follows the stated guidelines to ensure full credit for your work.*

**Instructions:**

- The work in this lab must be completed alone and must be your own.
- **Download the starter code files from the LAB3 Assignment on Canvas. Do not change the function names or given started code on your script.**
- A doctest is provided as an example of code functionality. Getting the same result as the doctest does not guarantee full credit. You are responsible for debugging and testing your code with enough data, you can share ideas and testing code during your recitation class. As a reminder, Gradescope should not be used to debug and test code!
- Each function must return the output (Do not use print in your final submission, otherwise your submissions will receive a -1 point deduction)
- **Do not include test code outside any function in the upload. Printing unwanted or ill-formatted data to output will cause the test cases to fail. Remove all your testing code before uploading your file (You can also remove the doctest). Do not include the input() function in your submission.**

**Goal:**

**[5 pts]** Write the function *isPalindrome*(*txt*). This function takes a string as a parameter and **returns** the Boolean value True if the string is a palindrome, False otherwise.

- A string is a palindrome if it is spelled the same both forward and backward.
- Sentences can also be palindromes, therefore, punctuation, capitalization, and spaces should be ignored. Numbers should not be ignored
- If the user provides an input that is not a string, program should return the Boolean value False

**[5 pts]** In the Module 2 video lectures, we discussed the Software Development Life Cycle and that testing your code is the foundation of solid software development. Getting used to writing testing code and running this code in parallel is now considered a good programming habit. We also discussed the advantages of unit testing, and how we can write down several test cases and let them be checked every time we make changes to our code. Write the test.py script to perform your testing for your *isPalindrome* function using the unittest module and prove that it works correctly. Remember to be as descriptive as possible and write as many cases as necessary (check the Hands On video for examples of a description). Unittest is intended to run test cases in bulk, so it should contain enough test cases to prove your code works. Your script should contain at least 6 cases with 3 tests per case. Examples of cases are testing words, lowercase, uppercase, mixed letters, sentences, strings with punctuation, spaces, input that is not a string, etc. You can discuss, suggest and share possible edge cases on Piazza or during your recitation class. Do not include the doctest examples in your unittest file

*** You will not receive credit for this assignment if you don't submit both pieces of code.*

*List of palindromes*:
http://www.palindromelist.net/

**Deliverables:**

- Submit your *isPalindrome* code in a file named LAB3.py to the Lab3 **GradeScope** assignment before the due date
- Submit your unittest code in a file named test.py to the Lab3 **CANVAS** assignment before the due date