**IVY TECH COMMUNITY COLLEGE**

**SDEV 220 Python Programming**

**Chapter 8**

**YOUR NAME: \_\_\_\_\_Giovanna Gorski\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**TO START YOUR PROGRAM**

At the top of each of your Python programs, you should have four lines of documentation:

Program name (the Python file name), Author (your name), Date last updated, and Purpose (a brief description of what the program accomplishes). Here is an example:

# Program name: TemperatureConverter.py

# Author: Carl LaFong

# Date last updated: 5/16/2012

# Purpose: Print out your name, address, phone #, and college major

Leave a blank line and then begin to write your program

**ASSIGNMENT: Programming Exercises #6 on page 367 with the specifications listed below.**

1. Programming Exercise #6 (Average Number of Words): The user will enter the name of the text file to be read.
   1. Name your Python program **AverageWordCount.py**
   2. There are two text files provided with this lab assignment, **text.txt** and **jabberwocky.txt**. Copy these two files into the same folder where you create your Python program.
   3. Your program should have a main function that asks the user for the name of the file to be opened, opens the text file, reads the contents into a list, closes the file, then reads each line to count lines and words.
   4. Hint: Use counters for the number of lines and the number of words. Each time you start a new line, increment both the line counter and the word counter by 1. Each time you encounter a space, increment the word counter again by 1.
   5. Display the number of lines read, the number of words read, and the average number of words per line. Format the average words per line figure to display with 1 decimal place.
   6. Be sure to include the four lines of documentation at the top of each program, and a short comment above every line or small section of Python code.

**ASSIGNMENT: Security in coding in general and specifically in Python**

1. Security is becoming a major part of the programming code. Go to some of the following websites. These will get you started. Read about secure coding and coding in Python. Search the Web and find additional information about the importance of writing secure code in today’s world. Write a summary of the information you learned about secure coding. Add it to the end of your lab assignment document. Be sure to cite your sources.
2. <https://mail.python.org/pipermail/tutor/2009-October/072150.html>
3. <http://python.about.com/od/cgiformswithpython/ss/ProgramSecurity.htm>
4. http://seclists.org/securecoding/2010/q3/6

To ensure that the code being created is secure, it is interesting to follow some defensive programming techniques. According to the sites listed, one important point to notice: under no circumstances should the application accept incorrect data entries. As a consequence, there must be a very strict control, avoiding unexpected behaviors, errors and wrong output. In addition to detecting an error, we need to know what to do if an unexpected output or error happen.

Programming languages such as Python have specific blocks of error handling (try / except blocks), which should be used in critical data entry situations.

In addition, my professor of Operating Systems in Brazil always said that a variable must be initialized, you never know what data it will hold in the computer’s memory. A badly initialized variable can be considered as a security breach in your program.

In conclusion, the articles listed demonstrate the importance of validating input data and other good programming practices to insure security at developed programs.

**HOW TO SUBMIT YOUR ASSIGNMENT**

Save and close your Python files. In Blackboard, go to the Class Session where this assignment is posted, attach your Python file(s), (Example: jcSalesPrediction.py) and submit for grading. For this assignment, you should submit two program files. You do not need to send the textfile.