CMSC 135 Introduction to Scripting

Programming Assignment 5

Scope:

In this lab, students will write and run Python Script.

Equipment/ Tool Needed:

This lab requires the use of the following:

* Laptop / Desktop
* Any version of Windows
* Any Web-Browser
* IDLE IDE

Lab Outcomes:

Upon completion of this lab, students will be able to:

* Understand basic structures and algorithms.
* Properly apply basic programming constructs and concepts including:
  + Variables and types (int, float, char, etc.)
  + Strings, arrays, structures
  + Sequential execution
  + Assignments (=, +,-,\*,/,% , etc.)
  + Functions, procedures, and calls
  + Debugging techniques
* Run Python Scripts

Steps:

* The following steps will be completed and submitted via a Microsoft word file along with Python source code file. Any other format will not be accepted and will result in a zero grade. Programming Labs are considered **incomplete**, if they do not compile and will result in a zero grade.

1. Check System Requirements
   1. Check the system requirement to know whether your computer supports IDLE.
   2. Download and Install IDLE (Refer to Python Installation Lab)
2. **Assignment 1:  Average of Numbers**

Design and write a program that calculates the average of all the numbers stored in the file. Assume a file containing a series of integers is named numbers.txt and exists on the computer disk.

(Hint: create numbers.txt file with series of integers for this assignment)

Program Output Sample:

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Shape

Description automatically generated with medium confidence

* 1. Use the following Specifications:

1. Assume the user will enter valid data (Integer/Float).
2. The program should round the averages to a maximum of two decimal places.
3. **Assignment 2: Exception Handling**

Modify the program that you wrote for Assignment 1 above, so it handles exceptions raised due to IO Error and any Value Error.

Console Output:

Example 1:

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

* 1. Use the following specifications:

1. It should handle any IO Error exceptions that are raised when the file is opened, and data is read from it.
2. It should handle any Value Error exceptions that are raised when the items that are read from the file are converted to a number.

Submission Requirements:

1. Submit
   1. LastNameFirstNameAssignment5\_1.py
   2. Word Document with algorithm/Pseudocode and screenshot of the output
   3. numbers.txt file
   4. You can create a zip file to submit all three filesLastNameFirstNamePLab5\_1.py
   5. Word Document with Algorithm/Pseudocode and screenshot of program output for both assignments. Label all screenshots.
   6. You should include one block comment at the top of each program containing the course name, the assignment/project number, your name, the date, and the pseudocode that you used to develop the assignment/project as listed below

#################################################################

# Class: CMSC135

# Instructor: Last Name First Name

# Program Assignment: 5\_1

# Program Name:     LastNameFirstNameAssignment1.py

# Author:             Student LastName FirstName

# Due Date:               01/22/2022

# Description: Give a brief description of each Program

#     I pledge that I have completed the programming assignment independently.

# I have not copied the code from a student or any source.

# I have not given my code to any student.

# Print your Name here: \_\_\_\_\_\_\_\_\_\_

# Pseudocode: Write Pseudocode here for the program

#

#

#

#

###################################################################

1. Submit

a. LastNameFirstNameAssignment5\_2.py

b. Word Document with algorithm/Pseudocode and screenshot of output

c. numbers.txt file

d. You can create a zip file to submit all three filesLastNameFirstNamePLab5\_2.py

e. Word Document with Algorithm/Pseudocode and screenshot of program output for both assignments. Label all screenshots.

f. You should include one block comment at the top of each program containing the course name, the assignment/project number, your name, the date, and the pseudocode that you used to develop the assignment/project as listed below

#################################################################

# Class: CMSC135

# Instructor: Last Name First Name

# Program Assignment: 5\_2

# Program Name:     Assignment1.py

# Author:             Student LastName FirstName

# Due Date:               01/22/2022

# Description: Give a brief description of each Program

#     I pledge that I have completed the programming assignment independently.

# I have not copied the code from a student or any source.

# I have not given my code to any student.

# Print your Name here: \_\_\_\_\_\_\_\_\_\_

# Pseudocode: Write Pseudocode here for the program

#

#

#

#

###################################################################

Evaluation: 100 Points. All requirements are graded. Use the grading criteria as your checklist to ensure you have completed all the requirements.

Academic Honesty in CMSC 135:

All students are expected to do their own work.  You may receive insights, editing and debugging help from tutors, fellow students, acquaintances, and the internet, but *you cannot share code.*  **You are expected to do all homework assignments by yourself – DO NOT GIVE YOUR CODE TO YOUR FRIENDS!** You are responsible for completing the assignments yourself.  (Your fingers are the only ones that should touch the keyboard).  You are not allowed to cut-and-paste code from other sources.  If your assignment uses features of the language that have not yet been covered in this course, you will be asked to defend your work.

The department sends all student submissions for each assignment to a program called **MOSS**, which tests for programming similarity.  This program is more than a simple “diff” application; it is an Artificial-Intelligence-aided comparison of your code to all other students’ code on the project.  It returns a measure of percent similarity.

**If you turn in the same assignment or uncommonly similar assignment to another student (past or present), or if your assignment is uncommonly similar to code found on the internet, the following policy will be applied to your project:**

**1. First time high match with other students or sources, grade 0% for project. .  
2. Second time high match with other students or sources, 0% for project, F for the course,** notify department chair, dean and file an academic dishonesty report.

Academic dishonesty will **not** be tolerated.

|  |  |
| --- | --- |
| Grading Criteria | Points |
| 1. File format requirements | 5 |
| 1. Word Document with the following requirements listed below: 2. Algorithm/Pseudocode. (-10) 3. Submit 3 Screen prints of the program run. (-10) | 20 |
| 1. Submit the LastNameFirstNamePLab5\_1.py file, and Submit the LastNameFirstNamePLab5\_2.py file. 2. Program passes sample data tests. (-10) 3. Program passes instructor data tests. (-10) 4. Quality of detailed steps and screenshots as per requirements listed above. (-5) 5. Incorrect use of indentation, naming convention, comments etc. (-5) 6. Not meet input requirement(s)/Took input incorrectly (-10) 7. Not display or produce all required output. (-10) 8. Not meet output requirements. (-10) | 70 |
| 1. Correct spelling & grammar. | 2.5 |
| 1. Font consistency | 2.5 |
| 1. Submitted the deliverables as per the due dates specified in the course schedule. Include a block of comments as per requirement. | 5 |

Questions:

It is up to each student to clarify any part of these guidelines, which may be unclear immediately after this assignment has been assigned.