**Assignment 1 – Basic Calculations**

**Dean Zeller 20 points**

### CG120 Due: Friday, August 31st

**Fall 2018**

***Objective***The student will calculate the sum and average on integer input, outputting the results.

***Background***: **Overall Program Design**

This assignment sets the stage for all future programs in class. There will be many assignments focesed on data processing. No matter how complex the processing will get, programs dealing with data analysis will generally follow three phases.

Phase 1: Gather input

Phase 2: Perform analysis

Phase 3: Output results

For this assignment, all three phases are quite simple. User input will be exactly three integers, the analysis will consist of just a sum and average, and the output does not require formatting. All three of these aspects will be covered in detail throughout the semester. For now, simply observe the overall program structure.

***Test Run:***

The following is a test run of the program you are going to create for this assignment. Any space text in **boldface** is data entered by the user.

Welcome to Assignment 1, Basic Calculations

Please enter your name => **Dean**

Hello, Dean

Introduction:

This program will calculate basic statistics on three integer numbers. It will run in three phases:

Phase 1 – Gather input from user

Phase 2 – Calculate sum and average

Phase 3 – Output results

Beginning Phase 1: Gather user input

-------------------------------------

Please enter three numbers. Only enter integer values, as error checking has not yet been implemented.

First number => **12**

Second number => **2**

Third number => **8**

Numbers entered: 12 2 8

Phase 1 complete

Beginning Phase 2: Performing calculations

------------------------------------------

Calculated sum

Calculated average

Phase 2 complete

Beginning Phase 3: Output table

--------------------------------

Data: 12 2 8

Sum: 22

Average: 7.333

Phase 3 complete

Exiting program

***Output Notes:***

1. Most lines above are generated by print statements.
2. Any line with **boldface** text indicates an input statement.
3. Observe the use of user instructions during execution. Follow this concept of giving instruction and feedback for all course programming.
4. Use a prompt of “=>” any time input is requested, without this prompt it is unclear whether the program is waiting or processing.
5. Indentation and blank lines are important, implement all indents and blank lines in both print and input statements, this will be important for organization in future programs.

***Statements:***

You will program a series of Python statements to replicate the test run. The output should look exactly like the test run above. To do this you will only need the following statements:

Documentation #This is documentation

Printing print(“Hello World!!!”)

Input(String) name = input(“Please enter you name:”)

Input(Integer) number = int(input(“Please enter a number:”))

Perform Calculation total = number1 + number2

***Instructions***

1. Open the IDLE shell window.
2. Create new program file named assignment1.py
3. Enter the appropriate block documentation to identify you as the primary programmer. See below for a block documentation template.
4. Save and execute your file. At this point it should do nothing as it is only documentation and not code.
5. All successive steps involve code to implement lines in the test run. After each step, test and execute your code, fixing any errors before moving to the next step.
6. Output lines 1 and 2 are simple print statements. Print a single string surrounded by “quotation” marks. You can include multiple strings, separate by a comma (,) or a plus sign (+).
7. In output line 3 the program is requesting string input from the user. This uses the input statement, discussed above. Within the input statement, use the appropriate prompt, as shown in the test output.
8. Following the name input, the next several lines are again simple print statements. Complete the introduction section, and test your code.
9. Within phase 1, the program accepts integer input from the user. Create three integer variables and accept user input for their initial values. You do not need to perform error-checking yet, though it will be introduced later in the course. See the input statement above on how to input integers.
10. The next line poses a unique problem, where both strings and variables are printed. Multiple items may be printed, separate by commas.
11. Phase 2 contains all the calculations to be performed. For this assignment, only the sum and average are calculated. Use the common mathematical operators to create the variables sum and average and calculate them appropriately. See the reference on calculations above.
12. Phase 3 contains the code to output the data and results. Use print statements to print the results, exactly as shown in the test run.
13. Upon completion, your program should execute exactly like the program test run. Test your program several times with different integer values, to make sure your program works correctly.
14. Use screencastomatic.com to record a 2-minute screen capture video of your program demo, with the following elements.
    1. Introduce yourself, and greet the teachers, TA, and students.
    2. Scroll through your code, making note of the assignment requirements and how you solved them.
    3. Run the test data used on the assignment write-up.
    4. Run the program two more times with other test data, and ensure it works correctly.
    5. Thank your audience and indicate that the video is complete, have some sort of fun ending message.
15. Create a zip file with your program file and screen capture video. Make sure all necessary files are in the submission, and no unnecessary files. Submit the zip file to the appropriate dropbox on canvas by the due date.

***Grading***

You will be graded on the following criteria:

*Effort* Creating the code to calculate the sum and average.

*Readability* Following the block documentation and line comments in the templates, and correctly naming the files

***Evaluation Instructions***

On Monday, the solution and all student submissions will be made available for download. Find three students in class to evaluate your work, and you are to evaluate their work. Use the rubric copies on the next page.

***Assignment 1 Grading Rubric Programmer Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

***CG120 Evaluated by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

Program Testing

\_\_\_\_\_\_ Execution

Does the program execute correctly?

-20 for inexecutable code

\_\_\_\_\_\_ Output

Does the video correctly demonstrate the execution of the program?

-20 for altered or fake output

\_\_\_\_\_\_ Test input/output

Does the program work correctly for the given test input from the assignment write-up?

-2 to -10 for errors in output

\_\_\_\_\_\_ Other input/output

Test the code for other combinations of data, does it work correctly?

-2 to -10 for insufficient testing

\_\_\_\_\_\_ Is the output formatted exactly like the test run?

-2 to -5 for incorrect formatting

-2 to -10 for elements missing

Submission Requirements

\_\_\_\_\_\_ Program

Does the submission have all required files to execute?

-10 for missing any necessary program file(s)

\_\_\_\_\_\_ Video

Is there a video demonstration included in the submission?

-10 for missing video

-2 to -9 for insufficient video or missing elements

\_\_\_\_\_\_ Program Style

Does the program follow the specified style guidelines including indentation, variable names, and blank spaces? -2 to -5

Documentation Requirements

\_\_\_\_\_\_ File documentation

Is there correct documentation at the top of the file following the documentation guidelines in the writeup.

-10 for no documentation

-2 to -9 for insufficient or incorrect documentation

\_\_\_\_\_\_ Spelling/grammar

Is the documentation properly written and free of spelling and grammar errors?

-1 to -5 for spelling and grammar errors