**Coding Exam Final**

This is a coding exam to test your skills at coding with access to any material of your choosing. You can use any IDE (Spyder, PyCharm etc.) of your choice. You can refer to your notes or browse the Internet and search for ideas/solutions/code snippets as in real life. Reuse as much code as you want. But you should **NOT** get help from others in or out of class using any communication medium. **Make sure you read the instructions carefully.**

**Scenario:** After Freshman Year you are given an Internship to automate some computations as part of a larger group. You are to provide your solution using the Python language.

You are provided a template file pre-named ***Custom\_Script.py*** that you should use so that all code by other interns can be integrated into one complete package. Hence, you must stick to the INPUT, OUTPUT format dictated by the template. You will be given a sample test file ***test\_data.csv*** to help you validate your code, but your evaluation will be based off a larger test data set that you do not have access to. *It should be your intent to make sure your script will handle all possible test cases as per specification*.

The python file you submit must have the name ***Custom\_Script.py*** as provided to you.This file is provided as a template along with stub code to read ***test\_data.csv*** and provide variables **INPUTS** & **OUTPUTS** for checking your code. **DO NOT CHANGE THESE NAMES**. Assume both script and data files are in the same directory.

1. **Part 0 Section of *Custom\_Script.py*: File Read Template Provided**

The file read routine is provided in the template ***Custom\_Script.py*** to read ***test\_data.csv***. It provides two lists INPUTS & OUTPUTS that you can use to check your function. Run ***Custom\_Script.py*** before you start coding to make sure.

***test\_data.csv*** has the format with a header as shown below.

|  |  |
| --- | --- |
| **INPUT** | **OUTPUT** |
| Input1 | Output1 |
| Input2 | Output2 |
| Input3 | Output3 |
| **.** | **.** |
| **.** | **.** |

**Your solution should be in two parts**:

1. **Part 1 Section of Custom\_Script.py: [50%]**

**Write a function** inside ***Custom\_Script.py*** that implements the following logic. **Your function should take as input one number and return as output one number.**

**The problem**: Given a measurement from a sensor ‘***input***’ you must write python code that computes the following ***output*** based on specific conditions and return the computed value:

1. If ***input*** less than 0 then return ***output*** as cosine of the input
2. If ***input*** equal to 0 then return ***output*** as **0**
3. If ***input*** greater than 0 then return ***output*** as square of sine of input

Sine and Cosine functions are available in the **math library**.

Assume all ***inputs*** are numerical values (Floats).

1. **Part 2 Section of Custom\_Script.py: [50%]**

**Write code to test your function** by calling it for **every value** in INPUTS (**one at a time**) and comparing it to the corresponding values in OUTPUTS.

Ensure you are comparing floating point numbers using a tolerance of 10-4. Ensure you can test against unknown number of rows of data. You do not know the length of INPUTS and OUTPUTS. **We will use a test file with different number of rows to test your solution**. You can assume INPUTS and OUTPUTS are of the same length and row values of INPUTS and OUTPUTS correspond to each other.

Output the number of successes based on the data given in ***test\_data.csv***. For example, if you get all inputs processed by your function to correctly match the OUTPUTS provided you should score 100%. Print to console proportion of success as: **Success = 100%**. If you only get 60% correct, then you should compute **Success = 60%**. This is your grade.

1. **Part 3 Section of Custom\_Script.py: Bonus Points (over 100%) [50%]**
2. Produce a line plot of INPUTS (X-Axis) versus OUTPUTS (Y-Axis) using the matplotlib library. Label your plot with X-Axis, Y-Axis Labels and title “INPUTS Vs OUTPUTS”.
3. Over plot YOUR function’s Outputs as points on the above line plot.

UPLOAD YOUR Solution ***Custom\_Script.py*** to eCampus.

Also enter your grade (Success= yourScore%) to eCampus in the Text Box of the link provided