**ENGR 102**

**Lab Assignment #2b [100 POINTS]**

**Activity**: Assignment: Writing your Programs – **To Do individually**, in lab or outside

You are to write the following 3 programs (though one of them has 3 versions), each of which should be done individually. Once the programs are written, you should create a single .zip file of the 5 programs and submit them.

**Program 1: [20 POINTS]**

Begin with Program 1 from Activity of last week’s Lab (1b). You are to convert that program to a new program that produces identical output. However, for all of the calculations, you are to instead create variables for all values that are either constants or are values that might vary in the calculation. As an example, if you previously had a line like: (***This will teach you reuse of code –smart strategy***)

print(3+2)

you would want to change it to a sequence of assignments such as:

a = 3

b = 2

c = a+b

print(c)

Please note the following:

1. Your print statements should each print just a single variable.
2. You should pick good names for your variables.
3. You do not have to perform the entire computation in one line; you can use multiple lines to perform the computation if you want.
4. It is OK to introduce variables to hold values that are not a “final” value. For example, if you were computing the area of a circle, you might store the radius in one variable, then the radius squared in another variable, and then later multiply that by pi to compute the area.

As a reminder, your program was to print the following 10 lines (the first 2 don’t change from before):

1. Your name, UIN, and section number of ENGR 102 that you are enrolled in
2. A sentence giving some interesting fact about yourself
3. The voltage across a conductor with resistance 20 and a current of 5.
4. The kinetic energy of an object with mass 100 and velocity 21
5. The Reynolds number for a fluid with velocity 100 and kinematic viscosity 1.2, with characteristic linear dimension 2.5.
6. The energy radiated per unit surface area (across all wavelengths) for a black body with temperature 2200. Use 5.67 x 10-8 for the Stefan-Boltzmann constant.
7. The production of a well after 20 days, if it had an initial production rate of 100, an initial decline rate of 2/day, and a hyperbolic constant of 0.8.
8. The average length of an M/M/1 queue with an arrival rate of 20 and a service rate of 35.
9. The shear stress when a normal stress of 20 is applied to a material with cohesion 2 and angle of internal friction 35 degrees
10. The scattering angle for maximum interference for light of wavelength 7.5 x 10-7 hitting a crystalline lattice with planes separated by a distance 1 x 10-6.

**Program 2: [This will help you understand cut & paste code reuse] [30 POINTS]**

In the earlier team project, your team put together a program that interpolated between two values. This was a one dimensional (1D) interpolation, since you were interpolating only a single value, the distance on the track. You are now going to extend that program to one that will linearly interpolate between two points in 3D.

1. Write a program that will take two observed 3D positions at two points in time, and then will calculate the 3D position at a third point in time. You should output the x, y, and z values for that position on separate lines. Begin by identifying the variables you will use, the names for those variables, and the computations that should occur for those variables. Then, write a program that will output the 3D position of the interpolated point on 3 separate lines. Save this program as Program2a.

For this initial program, you can use the following data values:

At time 13, observed position was (1, 3, 7)

At time 84, observed position was (23, -5, 10)

You want to find the position at time 50

1. Now, copy Program2a into a new program, Program2b. You are going to modify the program in the following ways:
   1. When outputting the position, follow the output by a line of dashes (“------------------“).
   2. Instead of just computing the interpolation at one point and printing the result, you will now compute it at 5 points. Copy the portion of the code (cut and paste the code) that is needed to recompute interpolation 5 times. You should now interpolate at the times in increments of 1 unit, starting at time 50 (i.e. at times 50, 51, 52, 53, 54), outputting the result each time. The line of dashes will separate each computation.
      1. Note: later we will see how we can do this more efficiently, without cutting-and-pasting code, but for now, cut-and-paste is fine.
2. Finally, copy Program2b into a new program, Program2c. Modify the program in the following way:
   1. Create variables for the starting time of interpolation, and the ending time of interpolation.
   2. You should display the results from interpolating at 5 points, evenly spaced from the beginning time to the ending time, inclusive.
   3. Experiment on your own with assigning different values to those variables verify that you are in fact interpolating correctly from one point to another.
   4. For the version you save and turn in, show an interpolation from time 20 to 50.

**Program 3: [50 POINTS]**

You are to create a program consisting of only the following lines of code. You may put these lines of code in any order, and can re-use the lines as frequently as you wish to. There will be more than one way to achieve the result – try to see if you can obtain the output using fewer lines of code.

x = 1

y = 10

z = 0

x = y

x += 1

y += x

y \*= x

z += x

z += y

print(z)

Your program should print out the following, when run:

1

3

11

28

123

100000000000000000000000000000000 *[Note: that’s 1032]*

4321

***HINT: The code could be simplified if you assume the previous result when computing the next one, in many cases. For instance, after the code to print 1, instead of resetting x to 1 and z to 0, you could just start with x having the value 1 and z having the value 1, and so with two more lines of z+=x, you have z=3.***