Attach the assignment to your HW/Lab ENGR 102 Section _______ **Lab 2a**
Date: _____ **DUE DATE: see website**
You name ____

ENGR 102 Sect #____ Lab 2a - team

100

Reading assignment:

Lecture Slides	L01- L02

Attention!!

For submission: pdf/word file and all py-files as asked in the assignment. If you do pictures by the phone –please make sure that we can read them. Do not submit multiple picturess, collect them all into one file (word or pdf). . You will be allowed to resubmit and reupload HW as many times as you want to within the due date/time, only last submission will be graded. No late submissions.

For submission you may use this file as a template: rename file including your name. Do not forget to put your name inside of this file as well.

If you are submitting py- files, make sure that they have a header. Use a team header for this assignment.

To do in a lab as a team

Use a team header and put all team members to all files.

What do you submit?

Lab2a_team#.pdf or word with all answers, derivations, screenshots, conclusions for activity 1 and activity 2. *TA's will assign the team number in the class.*

And any py-files you produced. You have to submit your py-files. Points will be taken off for missing py-files.

The purpose of this activity is to practice writing simple programs that require multiple variables and to ensure you understand the idea of interpolation. The individual assignment will build on this program.

You are to work together as a team to write a short program that performs the linear interpolation. Here is the scenario.

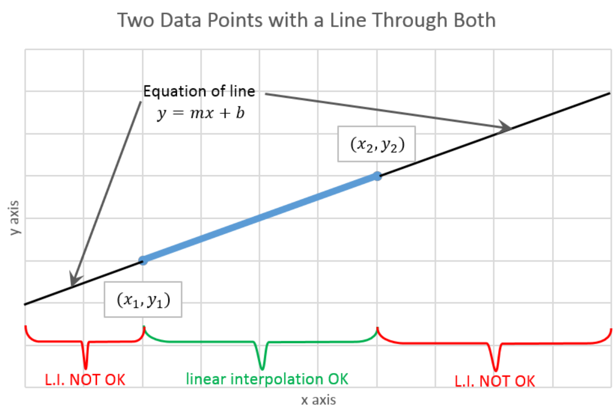
: Team Activity – Linear Interpolation Code

The purpose of this activity is to practice writing simple programs that require multiple variables, and to ensure you understand the idea **of interpolation**. One of the individual assignments in Lab Assignment 2b will build on this program.

You are to work together as a team to write a short program that performs linear interpolation

Please refer to the posted material on Linear Interpolation.

You name ____



From presentation:

- Solving for y, we have an expression for unknown y in terms of the (x, y) coordinates of Point 1 and Point 2 and in terms of the given value of x

$$y = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x - x_1) + y_1$$

- What variables do we need?
  - Let's use x_1 and y_1 for Point 1 location
  - Let's use x_2 and y_2 for Point 2 location
  - Let's use x_0 for the given x value of the point of interest
  - Let's use y_0 for the unknown y value that we seek
  - We may decide to use other variables, but this will get us going
- What steps do we take?
  - Assign values to x_1 and y_1
  - Assign values to x_2 and y_2
  - Assign x_0 its value

Calculate y_0 based on the expression developed above

- Let's make sure that we are performing Linear Interpolation not Linear Extrapolation
- To do that, check to make sure that x_0 is between x_1 and x_2 prior to performing the other calculations

How do we implement in Python?

- Now the program sequence looks like this:
  - Assign values to x_1 and y_1
  - Assign values to x_2 and y_2
  - Assign x_0 its value
  - If x_0 is between x_1 and x_2 (inclusive), do the following:
    - Calculate y_0 based on the expression we developed
    - Output or store the value of y_0
    - End program
  - If x_0 IS NOT between x_1 and x_2 (inclusive), do the following:
    - Print error message to screen
    - End program

Attach the assignment to your HW/Lab ENGR 102 Section _______ **Lab 2a**
Date: _____ **DUE DATE: see website**
You name ____

You are to <u>work together as a team</u> to write a short program that performs linear interpolation. Here is the scenario:

You are an engineer at NASA monitoring the International Space Station (ISS) as it orbits the Earth at a constant rate of speed. You want to be able to predict where the ISS is above the Earth at any point in time. To do this, you take a measurement of how far around the Earth the ISS has traveled at two points in time. Assume that NASA has very precise instruments for determining position. You note the time of the first position, and a short while later (before the ISS has completed one revolution), you take a second measurement for how far the ISS has traveled, again noting the time.

Now, it's your job to reconstruct the position of the ISS at any time between the first and second measurements. Since you assume the ISS is moving at a constant speed, this calculation can be found via linear interpolation. As a team, determine what variables you will need to use, and what formula(s) you will need to perform this calculation. *You should use variables for <u>all</u> of the values that could change.*

<u>Part 1. [50 pts]</u>

The first measurement was taken at time $t = 10$ minutes, and the second was taken 45 minutes later. At the first measurement, the ISS was 2,025 kilometers past Houston, TX. At the second measurement, the ISS was 23,025 kilometers past Houston.

- Write a program that determines, for any time between 10 and 55 minutes, where the ISS will be (in terms of kilometers past Houston). The time to evaluate at should be a variable in your program. The program should print both the time and the position at that time to the screen, with a line describing what is being output (<u>see example output below</u>). You should test your program at various times and make sure the results seem reasonable.
- For your final program that you submit, output the position at time of **25 minutes**. (Next week, we will see how you can read in numbers from a user, but for now, just assume it is a fixed number of minutes.)

<u>Questions to think about</u>: What happens if you enter $t = 0$ minutes as the time of interest? What is output as the position at that time? How do you interpret this result? Should the position at $t = 0$ minutes be at Houston? <u>Suggestion</u>: Hand draw a sketch of position versus time and plot the two known observations. Now, predict from the sketch what the calculated position will be for $t = 0$ minutes or for $t = 1500$ minutes.

Put your derivations, screen shots of your code, and outputs here

How to name a file? Something like that, for example HW2a_team#_ Act1. py

<u>Part 2. [50 pts]</u>

Now, let's make this a bit more interesting. The ISS orbits in a circle with **radius 6,745 kilometers**. Use the same observed data as before: **at 10 minutes**, the ISS is 2,025 kilometers past Houston, and **at 55** minutes, the ISS is 23,025 kilometers past Houston. Assume its speed is constant.

When a time is specified, we want to report the *distance from Houston*, not the total distance traveled. So, every time the ISS passes Houston, its "distance" from Houston gets reset to zero (0). So, if you go far into the future, say at a time of 5 hours, simple linear interpolation will not produce the result we want. You'll need to modify your code to report distances correctly regardless of the time.

You name ____

Here are a few hints for Part 2:

- If we use the same code from above and enter a time of 5 hours, we calculate a distance greater than the orbit's circumference. (Estimate that calculated distance from your plot.)
- However, we want to report a position of the ISS between 0 kilometers and the numerical value of the orbit's circumference expressed in kilometers.
- We could do this using a series of subtractions. We could perform successive subtractions of the circumference from the total position until the result was between 0 kilometers and the numerical value of circumference in kilometers. That would represent the position with respect to Houston.
- If we were clever, we could also use "modulo division" in Python. (Remember from Lecture 1?)
- Questions to think about: Is this linear "extrapolation"? If so, why are we are we using linear extrapolation despite all the warnings not to use it? Is there ever a case when using linear extrapolation is acceptable?
- Another Question to think about: Will the code for Part 2 output the correct answer for time (t) of 25 minutes as was used in Part 1?

Put your derivations, screen shots of your code, and outputs here
How to name a file? Something like that, for example HW2a_team#_ Act2.py
You can combine both act 1 and 2 into one py-file and name it as HW2a_team#_ Act1_2.py

Example output:
```
Part 1:
For t = 25 minutes, the position p = 9025.0 kilometers
Part 2:
For t = 5.0 hours, the position p = 10218.078642554414 kilometers
```

What do you submit?

- Lab2a_team#.pdf or word with **all answers, derivations, screenshots, conclusions** for activity 1 and 2.
- And any py-files you produced.