**ENGR 102 – Fall 2021**

**Lab Assignment #2a**

**Deliverables.**

Lab Assignment #2a consists of three team activities. Please submit the following files to Mimir.

* Lab2a\_Act1.pdf
* Lab2a\_Act2.pdf
* Lab2a\_Act3.py

Create the files for Activity #1 and Activity #2 using a word processing program such as Word or Google Docs, then save that in PDF format for submission. Please put the names of all team members that contributed at the top of each document. For Activity #3, be sure to include the team version of the header information at the top of your code.

**Activity #1**: Team Activity – Sequential Algorithm

This activity is meant to help illustrate the process of breaking more complex processes into sequential steps, and some of the choices and assumptions involved in doing so.

1. To begin, each member of your team should **individually** create a sequence of instructions for a person to get from your classroom in ZACH to the Student Recreation Center (on West Campus). Write your instructions as precisely as possible; “Go to the Rec” is not a good instruction, while “Turn left”, “Walk straight until you are in front of the H2O fountain”, “Stop after you pass the entrance to the MSC,” etc. are reasonable. It might help you to bring up [aggiemap.tamu.edu](https://aggiemap.tamu.edu/), for a map of campus.
2. Next, as a team, you should look at each member’s instructions one-by-one. Comment on each set of instructions, specifically noting whether the instructions are clear, whether they provide sufficient detail, and whether they would get someone to the destination. The person who wrote the instructions should not comment on their own instructions. **BEFORE MOVING ON:** Discuss as a team: Which one of the sets of instructions do you consider the “best”? Why?
3. **DO NOT READ UNTIL TASK 2 IS COMPLETE!**

As a team: Discuss together and answer the following questions. Your team should produce a **single PDF document** named Lab2a\_Act1.pdf with two items: (1) copies of each of the sets of instructions (you will need to share, e.g. by Google Drive, your instructions with each other), and (2) brief (a couple of sentences, or lists of no more than 10 things) answers to the following questions:

* 1. Which set of your team’s sequences of steps did you identify as being the best? Why?
  2. In what ways were the sets of sequences that were produced different?
  3. In what ways were the sets of sequences that were produced the same?
  4. Consider whether your choice of which of these would be the best set of instructions might change depending on the person following them. For example (you may think of other examples), would the best set change if:
     1. The person following them was already very familiar with campus, or had never set foot on campus.
     2. The person following the instructions was using a wheelchair, or the person following the instructions was interested in jogging.
     3. The weather was dark and raining outside, or it’s a beautiful and sunny 75 °F.

Briefly describe whether different sets of instructions might have been better options in other scenarios.

* 1. This was a very open-ended question. What questions might you have asked to begin with in order to better know how your sequential steps should have been written? The point here is to help you understand the importance of *requirements gathering* at the first stage of attacking a problem – **make sure you are solving the problem someone needs solved, rather than the one you want to solve**.

**Activity #2**: Team Activity – Program Analysis

This activity is also meant to illustrate the process of breaking a problem down into sequential steps and the beginnings of how that can become a computational procedure. As a team, you will read instructions to assemble a product, and then describe a list of steps that should be taken, sequentially, to obtain the result. You will not be writing a program (yet). Here are the details:

* Read the following instructions: <https://images.thdstatic.com/catalog/pdfImages/2e/2eb55cb6-57e9-457c-aef9-91d3c5196860.pdf>
* As a team, you are to write down the procedure to be followed as a series of individual steps. While the instructions already do this to some extent, we are looking for a more detailed and directly specified sequence to follow.
  + Each step should be a **single, small, discrete** operation. Multiple tasks or multiple checks should **not** be combined into one step.
  + Steps should describe a single action, such as “measure”, “check”, etc.
  + Examples (from a different task):
    - “Measure the mass of the empty container.”
    - “Calculate the distance traveled by multiplying velocity times time.”
    - “Check whether the two volume readings are within 5ml of each other.”
  + You do not need to try to describe how things are handled if a check is not passed (e.g. don’t use the word “if”). You can just use the term “check” if you are wanting to check a particular value (and it will be understood that something else is done if the check fails).
* Next, once you have completed a detailed instruction sequence, imagine that you would end up writing a program to help someone else perform this task at a larger scale (what if a customer wants to set up an assembly line?). The program may require data values to be entered and computed along the way, and these will be stored in variables.
  + You are to make a list of all the variables you would potentially use in this program.
  + For each variable, give the name that you would use for the variable in Python, along with a short description of what that variable is used to store. For example, you might specify a variable like this:
    - sphere1\_volume: Volume of first sphere

Your team should prepare a PDF document named Lab2a\_Act2.pdf giving the list of detailed instructions, along with the list of variables and their meaning.

**Activity #3**: 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. ***Please refer to the posted material on Linear Interpolation*.**

You are to work together as a team 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 all of the values that could change.*

**Part 1.**

The first measurement was taken at time 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 (see example output below). 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.)

Questions to think about: What happens if you enter 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 minutes be at Houston? Suggestion: 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 minutes or for minutes.

**Part 2.**

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.

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?

Please put the code for Parts 1 and 2 into one file named Lab2a\_Act3.py for submission to Mimir. Include the team header at the top of your code (see Canvas for the header information). Use a comment to clearly label the two parts.

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