ENGR 102 - Lab Assignment #2

Fall 2020

General course learning outcomes:

- demonstrate the use of basic programming techniques in the construction of computer programs, including techniques to collect, store, and manipulate data within a computer program.
- apply programming techniques to solve problems in engineering.
- complete a team programming assignment that ties together concepts learned in the class.

Activity 1: A Computational Procedure - to do in lab (team)

☑ *Discern important information from a technical article, then outline sequential steps required to use this information in an appropriate format.*
☑ *Resolve the required steps into a list of variables with useful variable names for a future hypothetical program.*

As a team, read the provided document about a testing procedure, and then prepare a PDF document for submission giving a list of detailed instructions and a list of variables and meanings as outlined below.

A) Read the provided document about concrete testing:
   a. As a team, write down the procedure to be followed as a series of individual steps. We are looking for a more detailed and directly specified sequence than the document provides.
   b. Each step should be a small, discrete operation. Multiple tasks or multiple checks should not be combined into one step.
   c. Steps should describe a single action, such as “measure”, “check”
      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”.
   d. In your description, do not use conditional statements (e.g., “‘if”). Write your instructions assuming that all checks will pass (do not account for failures).

B) Once you have completed a detailed instruction sequence, imagine that you must write a program to help the technician perform this task. The program will require several data values to be entered and computed along the way, each stored as a variable.
   a. Make a list of all variables you would potentially use in this program
   b. For each variable, give a name for the variable in Python, along with what that variable is used to store. For example, you might specify a variable like this:

      sphere1_volume: Volume of first sphere

(continued, next page)

Activity 2: A Program - to do in lab (team)

☑ *Create a list of appropriately named variables for a given task, write a computational program in Python, and provide output to the user.*
☑ *Review data interpolation.*

Write a short program that performs linear interpolation. This will require writing a simple program(s) that require multiple variables (and your individual assignment will build on this program).

Here is the scenario:

You arrive at a racetrack and observe a car moving around a track at a constant rate of speed. You would like to predict where the car is at any point in time. To do this, you take a measurement of how far around the track the vehicle has traveled at a certain point in time (assume that the track is well-marked so that you can determine position very precisely). You also note the time of this first position measurement. Before the vehicle passes the "starting" point on the track, you take a second measurement for how far around the track the vehicle has traveled, again noting the time. Now you'd like to reconstruct the position of the vehicle at any time between the first measurement and the second.

A) As a team, determine what variables you will need to use, and what formula(s) you will need to use to perform this calculation. You should use variables for all of the values that might change.

   hint, linear interpolation is: $y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0}$

B) Assume the first measurement was taken 30 seconds after you arrived, and the second was taken 45 seconds after you arrived. At the first measurement, the car was 50 meters past the starting line of the track, and at the second measurement, the car was 615 meters past the starting line.

   Your program should determine how many meters past the starting line the car will be for any time between 30 and 45 seconds. The evaluated time should be a program variable. The program should print the time *and* position to the screen, with text describing the output. Test your program at several times, and check for reasonable results (e.g., at 30s, 31s, 35s, 40s, 44s, and 45s).

C) For your submission, find the position at a time 37 seconds after you first arrived. (Next week we will see how to read numbers from a user, but for now assume it is a fixed number of seconds.)

Challenge (optional, for glory only):

Assume the racetrack is circular with radius 0.5 kilometers. Notice that every time the car passes the starting point of the track, its "distance" from the starting point gets reset to 0. So, if you go far into the future, say at a point 20 minutes after your arrival time, simple linear interpolation will not work. See if you can modify your code to report times correctly regardless of the time.

If your team completes the challenge, turn in your code, computing distance for times of *both* 37 seconds and 20 minutes.