DS2001 - CS Practicum

Spring 2023

Practicum 1 - Variables & Arithmetic Operators

Practicum is DUE at the end of your scheduled practicum.

We'll be working in pairs this week! With your partner, work on the programming assignment below. You may not finish every problem every week, but we expect your best effort.

We will let you know when there are 15 minutes left in class – at that point, it is time to wrap up your code and move on to the self-reflection portion of the assignment! Read the self-reflection questions below, write your answers, and save them in a PDF.

Submit both your code and your self-reflection PDF to our gradescope.

Grading Policy
You will receive full credit if you submit your self-reflection (with thoughtful answers) and any code by the deadline. All questions in the self-reflection must be answered for full credit.

Feedback
We will provide feedback on your code if requested in your self-reflection. Otherwise, we will not look at your code. Our feedback will mirror the expectations of your DS200 homeworks, and we will use the same criteria for grading your projects in DS2001. So, make sure you ask for feedback when helpful, and read our notes!

Today's Goals

- Create and modify variables
- Use arithmetic operators to solve problem
- Gather data from a file
- Report information to the user using Python's print

Working in Pairs

Each pair will work at one computer, and everyone contributes. Here's how we split up the work:

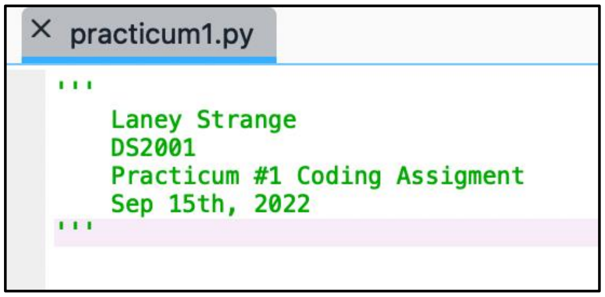
- Navigator: Dictates the code to be written. Explains the *why* as we go. Checks for syntax errors.
- Driver: Writes the code. Listens closely to the navigator. Asks questions whenever there is a lack of clarity.
- Both driver and navigator are responsible for contributing to the work, and for making space for the other person to make contributions.

Programming Assignment

0. **Getting Started**

Open up Anaconda and click on Spyder.

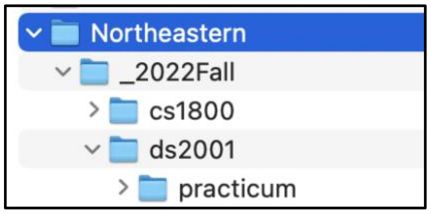
Create a new file by selecting File > New File from the menu, and name it practicum1.py. Replace the default stuff in the file with a header comment at the top that includes your name, the date, the class, and the practicum information. Like this:



```
'''
    Laney Strange
    DS2001
    Practicum #1 Coding Assigment
    Sep 15th, 2022
'''
```

Save the file in a folder that you create just for DS2001. Creating files and folders looks a little different in Windows vs Mac, and we are here to help if you need it!

For example, the file structure on Laney's computer looks like: Documents > Northeastern > 2022Fall > DS2001 > Practicum.



And that's where Laney would save her file practicum1.py. You don't need to have that many layers, but make sure you're organized for the semester.

Now we've got our Python file ready to go! For the rest of the assignment today, we'll write a Python program that works with numbers, strings, input, and output. Like all data science programs, we'll break our work into three parts: (1) gathering input, (2) computations, and (3) communication.

1. **Gathering Input**

Profs. Ab and Laney are both runners, and love to talk about running. All the time. So that's our theme for today -- in particular, marathons (which are 26.2 miles)!

Define a main function in your practicum1.py file. Above this definition, put the filename marathon.txt in a constant variable.

The data is marathons.txt is the following

- Line 1: Runner's name
- Line 2: Number of marathons they have run
- Line 3: How many minutes it took them to run their fastest marathon

Inside of main, use with/open/as (like you've seen in 2000 lecture) to read in the data in marathon.txt. Each line should be saved as a separate variable. Be sure to also add a comment that describes what this section of code does!

2. **Computations**

Start with a comment that describes what this section of your code will do.

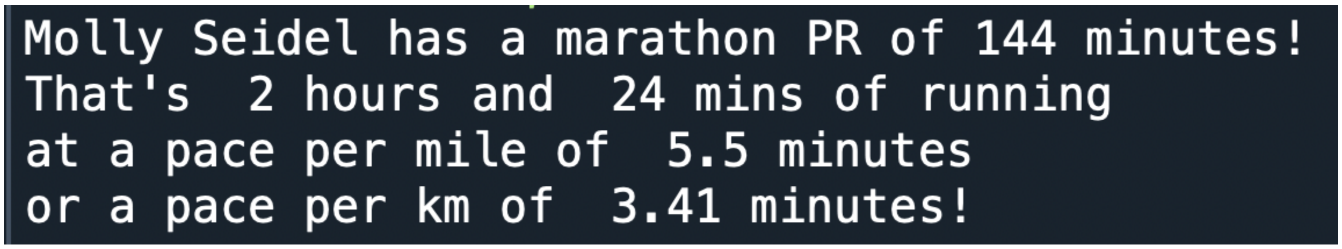
Use Python's arithmetic operators to figure out...

- The runner in marathon.txt's pace per mile in minutes (*xx.yy* minutes per mile) when they ran their fastest marathon.
- The number of kilometers in ONE marathon (there are 1.61 kilometers in a mile).
- The runner in marathon.txt's pace per kilometer in minutes (*xx.yy* minutes per kilometer) when they ran their fastest marathon.
- The total number of hours and minutes it took the runner to run their fastest marathon (*x* hours and *y* minutes).

3. **Communication**

Start with a comment that describes what this section of your code will do.

Use Python's print function to report back the runner's fastest marathon in minutes, minutes per mile for their fastest marathon rounded to 2 decimal places, minutes per kilometer for their fastest marathon rounded to 2 decimal places, and total hours/minutes for their fastest marathon. Here's an example:



```
Molly Seidel has a marathon PR of 144 minutes!
That's  2 hours and  24 mins of running
at a pace per mile of  5.5 minutes
or a pace per km of  3.41 minutes!
```

At this point, please complete the self-reflection! If you still have time afterwards, scroll down for a little extra/bonus work you can do on this assignment.

Self-Reflection

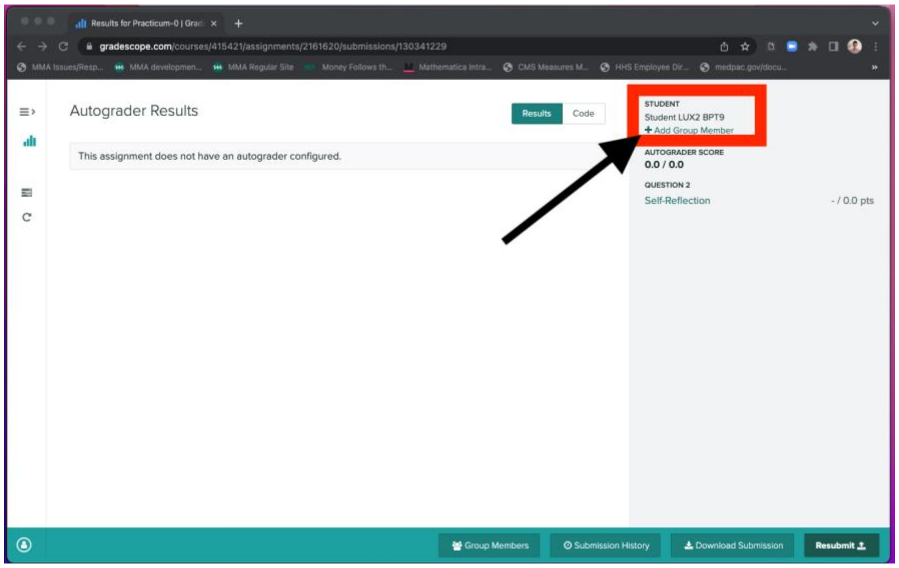
Create a PDF with answers to the following questions (each should be answered collaboratively with your partner unless instructed otherwise):

- Does this work reflect the best effort of both partners?
- In what way(s) did each partner (1) make contributions to the assignment, and (2) give the other person space to make contributions? (Each partner should respond to this question separately.)
- Which problem(s) would you like feedback on and why?

Submitting Your Work

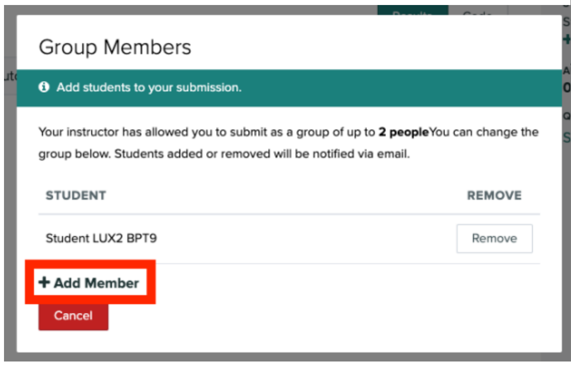
Only one person per group needs to submit, but that person needs to tag their partner to make sure everyone gets credit. Like last week, one person from each group submits your code and self-reflections on gradescope. Then, follow the steps below to add your partner to the submission:

1. After submitting the code and self-reflection, look at the top right corner of Gradescope. You will see your name or an anonymous name under STUDENT. Under that, click "+ Add Group Member":



This assignment does not have an autograder configured.

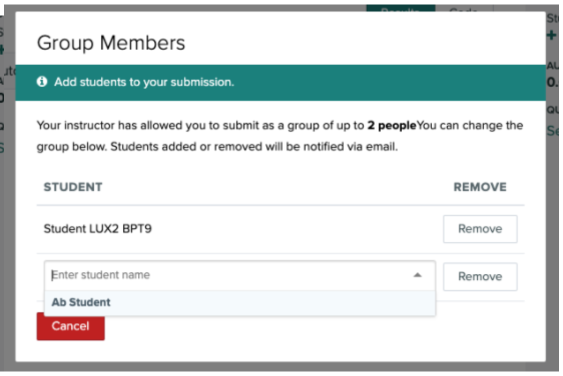
2. In the window that opens, click "+ Add Member", and then start typing your partner's name to find them in the class list. Finally, click "Save":


Group Members

Add students to your submission.

Your instructor has allowed you to submit as a group of up to **2 people**You can change the group below. Students added or removed will be notified via email.

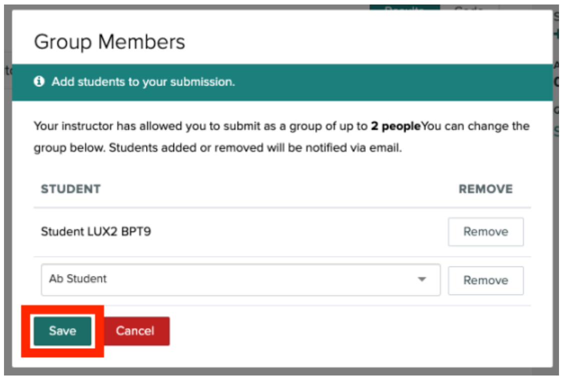
STUDENT	REMOVE
Student LUX2 BPT9	Remove


Group Members

Add students to your submission.

Your instructor has allowed you to submit as a group of up to **2 people**You can change the group below. Students added or removed will be notified via email.

STUDENT	REMOVE
Student LUX2 BPT9	Remove
Enter student name	Remove


Group Members

Add students to your submission.

Your instructor has allowed you to submit as a group of up to **2 people**You can change the group below. Students added or removed will be notified via email.

STUDENT	REMOVE
Student LUX2 BPT9	Remove
Ab Student	Remove

3. Your partner should receive an email notifying them that they were added to the assignment.
4. Now you are done! Congrats!

Keep Going (if time!)

Here are some ways you can improve and build on what we've done so far. Try any/all of them in the time we have left.

- Make your own text file with your marathon information, Ab's marathon information, or look up a pro's! Change the filename constant at the top of your program, and see if your code still works with this new file. If not, think about what you need to fix.
- Are your print outputs lined up nicely like in our example? If not, try using .strip() on string data from your txt file.
- Did you use the literal values 26.2 and 1.61 in your code? If so, try saving them in constants at the very top of your file/
- When *computing* the pace per kilometer, break it into minutes and seconds instead of *xx.yy* minutes. In our Molly example, 3.41 minutes works out to 3 minutes and 25 seconds. Write out a plan on paper before you jump into the code!
- The Olympic trials marathon in Atlanta was a 4-mile loop. Compute and report the number of times Molly would have run the full loop, plus how many miles to get up to 26.2?
- Let's say Molly did the first half of her PR marathon in 3 minutes and 40 seconds per kilometer. Compute and report the pace per kilometer she would have to average in the second half to make her overall average 3 minutes 25 seconds.