Hi,

The course contains two projects related to the concepts covered in the class. Projects will be implemented in Python. The implicit assumption is that students already know how to build python programs.

If you don't know or think you need a brief review of syntax and various language utilities. The following is the list of resources you can follow:

Attention: There will be an overlap of content covered in these tutorials, I would recommend you pick one resource for each topic and learn.

Installation:

https://realpython.com/installing-python/

Course:

1. https://developers.google.com/edu/python (Recommended, completing this course alone is more than enough to build the project.)

Introduction to Python / Syntax:

1. https://programiz.pro/learn/master-python
2. https://python.swaroopch.com/basics.html

   More resources:

   https://wiki.python.org/moin/BeginnersGuide/Programmers

Strings:

1. https://developers.google.com/edu/python/strings
2. https://www.datacamp.com/tutorial/python-string-tutorial

Data Structures:

1. Dictionary:
   a. https://realpython.com/python-dicts/
   b. https://www.dataquest.io/blog/python-dictionaries/
   c. https://www.learnpython.org/en/Dictionaries
2. Lists:
   a. https://www.dataquest.io/blog/tutorial-demystifying-python-lists/
   b. https://developers.google.com/edu/python/lists
3. Stack:
   a. https://realpython.com/how-to-implement-python-stack/

4. Trees:
    a. https://paigeshin1991.medium.com/understanding-tree-data-structures-in-python-e249802d775e
    b. https://builtin.com/articles/tree-python

Recursion:

1. https://www.youtube.com/watch?v=wMNrSM5RFMc
2. https://www.codecademy.com/learn/learn-recursion-python/modules/recursion-python/cheatsheet
3. https://realpython.com/python-recursion/

Debugging:

1. https://realpython.com/python-debugging-pdb/
2. https://medium.com/@yellalena/a-guide-to-debugging-python-code-and-why-you-should-learn-it-ae30d20419b7

Note:

You have about one month to learn these resources before we release project 1. I strongly recommend you go through these. I also want to make some things clear about the project. Since it is a 400-level course, we assume that you are proficient in the language and how to fix compilation errors and debug the bugs in your code. Questions such as "This is my code, and I don't know how to fix the issue, can you do it for me" will not receive any response. Because there are 400 students in this class, providing a correct answer to the above question will take approximately 30 minutes. In total, it will take around 8 days (400 x 30 minutes) for a single project. This is not practically feasible. We will only help you with conceptual understanding of the project and provide guidance on how to design and various ways to implement the project, we will **not debug** your program.

Questions like the following will receive a response:

1. What is the correct way to distinguish between ints and floats in the parser?
2. Can you help me to understand the tradeoff between possible approaches to solve the problem?
3. What are the issues in tokening the entire program at once and building a parse tree VS combining both Lexer and parser on a per token basis?

All the best for your projects!