Intermediate Python class syllabus

Course title	Intermediate Python
Prerequisites	Passing grade in Introduction to Programming with Python.
Schedule	Monday (in person) and Thursday (remote) 6:00pm to 7:30pm.
Course length	15 weeks

Overview

This is a continuation of the Introduction to Programming with Python class, and we'll continue where we left off in that class and finish learning about basic concepts in modern programming and dive deeper into Web Programming.

Objectives

The objective of this class is to reinforce your knowledge of programming by teaching you the basic of Web Programming. By the end of this class students should be able to write their own basic web applications using Python (Flask), HTML, and CSS.

Homework

Homework assignments will be assigned every week. Assignments should be submitted via Canvas. The assignment's due date and requirements will be detailed in Canvas as well.

Assessments and final exam

There will be an assessment at the end of every unit and a final exam at the end of the class.

Grade requirements

A minimum of 70% is required to pass this class. Your grade will be made up of a combination of the homework, unit assess, and final exam.

How can you succeed in this class?

- Do your homework
- Write the code yourself, no copy/paste
- Run all of the code that you write
- Take notes
- Use Slack
- Ask questions
- Work with your classmates

How can you fail in this class?

- You don't write code
- You don't take notes
- You don't ask questions

Outline *

Unit 1: Class Introduction

Beginning of the Python 2 class. We'll continue where we left off in Python 1, starting with a review of the main topics from Python 1: lists, loops, conditionals, and functions.

Unit 2: Dictionaries

Compare dictionaries to lists as a data structure for storing data. Compare uses and benefits of a dictionary vs. a list. When would someone reach for a dictionary as opposed to a list?

Unit 3: Introduction to Flask

Discuss imports, annotations. HTTP basics, like routing and methods, clients vs. servers. This unit should also include a review of HTML and CSS.

Unit 4: Classes

In this unit we will learn about classes in Python. We'll cover their basic functionality, the concept of encapsulation and when and why it's useful, and also talk about how we've already been using classes (Flash, int/string/float/etc.)

Unit 5: File IO

Learn about reading and writing to files in our computer. We'll also cover the basics of serialization and deserialization. What is the "state" of a program and how can we store it in a file that we later reload into memory. After learning about reading and writing state to files, we can discuss databases and their application in larger programs.

Unit 6: Error handling

Basics of error handling and why it's needed in program. Learn about errors resulting from user inputs vs. application errors. Learn about detecting and recovering from errors.

Unit 7: Modules

As projects grow in size and complexity, they are broken up into multiple files. In Python, each file of a program is module. We'll learn about how to create and work with our own modules.

* *The order in which units are covered may change depending on how the class is progressing.*