Introduction to Programming class syllabus

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

Overview

This is an introduction to computer programming with Python. No previous experience with programming is expected, you simply need access to a computer and to know the basics of how to use it. By the end of this class you will be able to write basic Python programs.

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.

In class exercises

Most class periods will involve in class exercises that you are expected to participate in and will be graded on.

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
- Participate in the in class exercises
- 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

Students will be introduced to the class, the tools we'll use, and to each other. We'll go over what is required in order to succeed. We'll also make sure everyone has Python and a text editor installed. Finally, we'll introduce students to the basics of computers, programs, and programming.

Unit 2: Variables, values, and user IO

We're now ready to start coding, so we'll begin by introducing the basic components of programming: variables, values and operators. We'll also learn how to write to the screen and read user input.

Unit 3: Conditionals

We'll be learning about conditionals which will allow us to write programs that perform different actions depending on a set of conditions that you specify. We'll take this as a chance to review boolean operations.

Unit 4: Lists and loops

Introduction to lists and loops. With this complexity comes new creative ways in which we can solve problems with programming.

Unit 5: Functions

In this unit we'll cover functions. Functions allow us to reuse our code and execute it whenever needed.

Unit 6: 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?

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