

# Causal 8607: Assignment 2

Sam Clark

2026-01-22

## Format and due date

Prepare this assignment by creating a Quarto markdown document and rendering it to **PDF** in the Positron integrated development environment (IDE). Upload your PDF document to the class CarmenCanvas website ‘Assignment 2’ by **23:59 on Thursday January 29**.

It is essential that you complete this assignment *before* class on January 29.

## Python object-oriented programming

Python is inherently an object-oriented language. Google’s AI summary is a nice succinct description of what we mean by ‘object-oriented’:

“Object-Oriented Programming (OOP) is a software design model that organizes code around objects, which bundle related data (attributes/properties) and behavior (methods/functions) together, modeling real-world entities, making complex software modular, reusable, and easier to maintain. Key principles include Encapsulation (hiding data), Inheritance (sharing traits between classes), Abstraction (simplifying usage), and Polymorphism (objects responding differently to the same command).”

The text we use for this class is written by a computer scientist who takes advantage of Python’s OOP features. For that reason, you will need to know at least the basics of how Python is written in OOP style. Following from last week, you will use Corey Shafer’s series of instructional videos that explain Python’s OOP features.

You are **required** to watch the series of videos below that cover the basic OOP features of Python. You will turn in a transcript of your work following along with the videos doing all the examples that they contain. Create a single Quarto document with sections for each video. Within each video’s section create code blocks, and if you want, text to demonstrate and explain what you learned from the video. Render the final document to create an PDF

document with your text and the results of running the code blocks. I want to see that you followed along, executed all the code, and understood what you were learning. You are free to organize your ‘proof of learning’ document as you wish. Remember that this video might be helpful as you get going on your Quarto document: [Your First Python Project in Positron](#).

### Required Python learning videos

- [Python OOP Tutorial 1: Classes and Instances](#). Time: 15:24. [Code snippets](#).
- [Python OOP Tutorial 2: Class Variables](#). Time: 11:41. [Code snippets](#).
- [Python OOP Tutorial 3: Class Methods and Static Methods](#). Time: 15:20. [Code snippets](#).
- [Python OOP Tutorial 4: Inheritance - Creating Subclasses](#). Time: 19:40. [Code snippets 1](#). [Code snippets 2](#).
- [Python OOP Tutorial 5: Special \(Magic/Dunder\) Methods](#). Time: 13:49. [Code snippets](#).
- [Python OOP Tutorial 6: Property Decorators - Getters, Setters, and Deleters](#). Time: 9:32. [Code snippets](#).

## Matplotlib

Matplotlib is a Python module for plotting, and obviously, you need to learn how to make plots in Python! You are **required** to watch the Corey Shafer video below that covers the basics of matplotlib. Add to your Quarto document transcript to cover what you learn and try out in this video.

- [Matplotlib Tutorial \(Part 1\): Creating and Customizing Our First Plots](#). Time: 35:00.

Like all major Python modules, matplotlib has its own website with all the information you will ever need to use it: <https://matplotlib.org/stable/users/index>.

## Plotnine

Plotnine is another Python plotting module that builds on the ‘grammar of graphics’ concept, similar to ggplot in R. This module is less developed but could be very useful for those who are familiar with and like ggplot. You are **required** to watch the video below that covers the basics of matplotlib. Add to your Quarto document transcript to cover what you learn and try out in this video.

- [Grammar of Graphics in Python with Plotnine - posit::conf\(2023\)](#).

Like matplotlib, plotnine has a website with all its documentationa: <https://plotnine.org/guide/introduction.html>.