Object-Oriented Programming

Overview of OOP

Object-oriented programming (OOP) is a programming paradigm that is often described as "modeling the real world" where information is encapsulated into a structured unit (an object) with a state (attributes of the object) and behavior (methods that can be used to work with the information). An object is created as an instance of a class, which is the blueprint for what a specific object, treated as a data type, is, outlining its state and behaviors. Each object, created by instantiating a class, has its own data; meaning that while the limits of their behaviors are the same, the attributes themselves are common but not shared between objects.

How OOP could fit into my teaching context

In AP CSA this year, I went through the sequence of units as provided in the CED by CollegeBoard, beginning with primitives, then using objects and methods, followed by conditionals and iteration, then writing full classes, leading into arrays, then finishing with inheritance and recursion. Reflecting and planning for next year, I'm considering moving things around a bit. Rather than a unit focused entirely on using objects and methods, it might make more sense to mix those elements and vocabulary into the rest of the material ranging from conditonals and iteration to arrays, since Strings are objects and slowly layering in the OOP concepts should make getting to writing full classes feel more natural. From there, inheritance could be an extension (pun intended) of the same paradigm and the whole thing should flow much more gracefully.

How I might teach OOP

In my Intro to Coding class, and perhaps in a future Game Design course, I plan to use something like Retro or Pygame to give a specific goal and context to OOP. I found building a video game using retro to lend itself well to the "modeling the real world" idea of OOP, where creating a player, enemies, even the projectiles flying around the game, all made sense in the context of "what is their state, what is their behavior", allowing me to focus more on what am I actually making, rather than what am I supposed to be doing. I would love more time to spend on these types of concepts, but without a dedicated course, focusing so much on OOP in such an intro class feels like too much too fast without a firm grasp on the basics.