Object-Oriented Programming

Overview of OOP

Object-Oriented Programming (OOP) is a way of programming centered around classes and objects. A class defines a new object ”type” that has certain properties (state) and possible actions (behavior). An object is a single instance of a class. Since a class is a general definition for objects of that class ”type,” there may be aspects of the state that are undefined or set to some default value. When an object is created, parts of its state may be defined/re-defined. The state and behavior of the objects created while a program runs determine the output of the program. Defining a class allows us to create many objects of the same type. OOP allows for the reuse of code to make many similar objects that behave in the same way. The keyword ”self” stores a reference to the object itself and allows for an object to access/change its own state and call its own methods on itself. When using OOP, objects will often have other objects as a part of their state. This allows for the interplay of many different behaviors and states between objects. In general, the state of an object influences its behavior and vice-versa. Some concepts that were not truly covered in this class that are important to this topic include inheritance and polymorphism. Additionally, there are many interesting implementations of OOP, including the state design pattern. OOP allows developers to design their programs in ways that sustainably reuse code and make it easier to add functionality to their programs.

How I could fit OOP into my anticipated teaching context

I believe that OOP is the logical next step for students to learn after a basic understanding of variables, functions, conditionals, and loops. I hope to teach computer science in a high school, either using Python or Java, and I hope to have a certain level of freedom. I want students to be excited by what they are programming, and I believe that OOP will allow students to create programs with the level of complexity required to intrigue students. OOP can also lend itself to collaborative programming, where each student works on the behavior of a single (few) class(es). The NYS standards that this topic aligns with are 9-12.CT.8 and 9-12.CT.10.

How I might teach OOP

I would introduce OOP after an introduction to the concepts of variables, functions, conditionals, and loops. I really like the idea of the games project we did in this course as a final project. I plan to give students a similar open-ended project that allows them to create something with programming that they find personally meaningful. One of my hopes as a teacher is that students are able to see the way they might use programming to develop something they personally **want** to create. I would try throughout the topic to create projects for students to work on that they find interesting and could see the practical use of.