

Object-Oriented Programming (OOP)

Concept overview

OOP is a certain way of programming that is based on the use of classes and objects. Classes are essentially blueprints for how objects of that class work. An individual object is an instance of that class. For example, if human was the class, all the humans in the world would be objects of the human class. I, Andrew, would be an individual object which would be an instance of the human class. Classes include all the data for how objects of that class work. These are in the form of attributes and methods. Attributes are variables contained in the objects that are shared amongst all instances of the class, and methods are functions contained in each of the instances themselves. All classes have the `__init__` method which is called whenever an instance of a class is created. This method includes a parameter `self`, which is not a given input but rather a variable that can be used to access that specific instance of the class. `__init__` also takes in any inputs expected when the instance is created. To create an instance of a class, you simply choose a variable and assign it the class name followed by parentheses encasing any inputs. This topic is important due to the usefulness of OOP. While functional programming is great for solving simple problems, OOP is better suited for more complex problems. Along with this, a project made with OOP is much easier to expand upon than one made with functional programming.

How OOP could fit into my teaching context

As a prospective math and computer science educator, I do not exactly have a teaching context yet. Thus, I chose to consider the New York State Computer Science and Digital Fluency Learning Standards. Standard 9-12.CT.4 says that students should be able to “[i]mplement a program using a combination of student-defined and third-party functions to organize the computation” (New York State Education Department, 2020). I believe I can use OOP to accomplish this goal because breaking things down into smaller parts that all work together is an essential aspect of OOP. Assigning a game project with a library like retro games that we used in this class could align nicely with this standard.

How I might teach OOP

This class was the first time I ever worked with OOP. Because of this, I had to do some additional research on what it was, along with asking friends questions. When trying to grasp the concept, what stuck with me most was the example of animals. Even in the Python documentation, they use the example of a class named `Dog` and they create instances of the class for different dogs, such as `Fido` and `Buddy`. Thus, to teach this concept I would use a similar basic example to the one that’s given in the Python documentation, but instead do it for a farm animal such as a pig. Then, I would have the students create other classes for cows, sheep, horses, etc. We could start off simple with just the ability to name an instance, but then get more complicated. One idea is adding the ability to feed an animal until it is full. What food you can feed an animal and how full an animal gets depends on the type of animal.

After the students have worked with OOP a bit, having them explore a more complicated example like the asteroid game we worked with in class could absolutely be beneficial. After exploring, having them use the knowledge they gained earlier to add to the game would also be great! The combination of having them work with simple code, and adding on to more

complicated code should help teach them the concepts of OOP. It should also show them how great OOP is when it comes to expansion.

References

- GeeksforGeeks. (2023, August 18). *Self in python class*. GeeksforGeeks.
<https://www.geeksforgeeks.org/self-in-python-class/>
- GeeksforGeeks. (2024a, March 15). *Python classes and objects*. GeeksforGeeks.
<https://www.geeksforgeeks.org/python-classes-and-objects/>
- GeeksforGeeks. (2024b, April 8). *Python oops concepts*. GeeksforGeeks.
<https://www.geeksforgeeks.org/python-oops-concepts/>
- New York State Education Department. (2020). Computer Science and digital fluency learning standards. <https://www.nysed.gov/sites/default/files/programs/curriculum-instruction/computer-science-digital-fluency-standards-k-12.pdf>
- Python Software Foundation. (n.d.). *9. classes*. Python documentation.
<https://docs.python.org/3/tutorial/classes.html>