Object Orientation Intro

Previously we...

- Wrote the code to work with the data we supply to it
- Had the code and data be pretty much separated
- We could have a function that modified the data somehow and also some associated data that we pass into it through the function call in the main function, but that was never expressed in the code itself

Why do we care if data and code are in the same place?

- Once things get complicated the approach of having data and code working with it as separate instances makes life of a programmer very difficult.
- This requires putting a lot of effort into how the programs are written (style guides and etc.)
- It is easy to introduce bugs because it is not always clear how the data and code are linked.

Main idea of object orientation

- You might have heard about different principles of OOP such as polymorphism or encapsulation, but these are essentially the consequences of a more general idea…
- We put the data and the code that processes it in the same logical instance in the program
- To do that, we need to specify how this instance operates, namely what data it needs to be created and how it processes this data
- We can also define hierarchy with regards to such instance specifications. We could define an “Animal” and say that “Cat” and “Dog” are derived from it.

Terminology

- Instance specification is called “Class”
- A particular instance is called an “Object”
- Data that each object of a particular class holds is called “Attributes”
- The functions to work with this data within the class are called “Methods”