# Exercise 1.5: Object-Oriented Programming in Python

## Learning Goals

- Apply object-oriented programming concepts to your Recipe app

## Reflection Questions

1. In your own words, what is object-oriented programming? What are the benefits of OOP?

   Object-oriented programming is when you have stored data and/or method attributes into objects that are called classes, which then come to define the characteristics of that class. The benefit here is that you can then generate other objects that contain the same attributes but that are empty and ready for inputting for further use.

2. What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work.

   Objects are structures that contain any data and/or procedural methods that are subsequently used in some way. Classes are the uniform internal structure of how that data/methods are organized. A real world example would be cars. You could say that cars are the class, because for the most part, they are structured and function in a similar way, and all the individual types of cars are the objects because they all have their own unique features as well. For example, all cars have tires, that is a component of the class, but each type of car will have a different kind of tire depending on its make, model, function, etc, which is part it its uniqueness as an object.

3. In your own words, write brief explanations of the following OOP concepts; 100 to 200 words per method is fine.

| Method | Description |
|---|---|
| Inheritance | Inheritance is when a child class inherits attributes from a parent class. This can only work one way, the parent class cannot inherit from the child. |
| Polymorphism | Polymorphism is when a data or method attribute has the same name but performs different functions depending on where it is defined. As long as it is defined separately and exclusively in one place, if should not conflict if you use it again somewhere else with a different input. |
| Operator Overloading | Operator overloading is when you define a function that Python already uses for your own custom class. Python operators won't work for custom classes, so you have to surround them with double underscores like so: __add__() |