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?

- This programming style uses objects as the name implies. Each object is structured from classes that holds its data and behaviors.

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

- Classes are a template for creating objects. Objects are instances of classes with each their own attributes.

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

Method	Description
Inheritance	It allows a new class to adopt the attributes and methods of an existing class, facilitating code reuse and the creation of hierarchical relationships. For example, if you have a Recipe class, you could create a BakingRecipe subclass that inherits from Recipe but also adds specific attributes or methods for baked goods.
Polymorphism	This concept allows objects to be treated as instances of their parent class rather than their actual class. It allows methods to be used in the child class that have the same name as methods in the parent class. In the recipe example, if Recipe has a method display, a subclass BakingRecipe could have its own version of display that's more specific to baking recipes. The right version of the method is chosen based on the object being used.
Operator Overloading	This allows operators in the language to be used in different ways depending on the data types of the operands. For example, the + operator can be used to add numbers, concatenate strings, or in the context of a Recipe class, combine two lists of ingredients.