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?
   Ans: OOP is like building with blocks. Instead of writing all the code in one go, you create individual "objects" that represent things and actions. These objects can be reused, modified easily, and help keep your code neat and organized, OOP makes coding more flexible and manageable.

2. What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work.
   Ans: In python, a class is a blueprint for creating objects. An object is an instance of a class, and it encapsulates data(attributes) and functions(methods) that operate on the data. For example, class can be 'car', attributes can be 'brand', 'model', 'color', and methods can be 'start_engine()', 'drive()', and 'stop()'.

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 like passing down skills in a family. A new thing(like a specific type of car) can learn and use the tricks of a more general thing(like any vehicle). It's a way to make code share and reuse its cool abilities.
Polymorphism	Polymorphism is like having a single remote control that works for many different TVs. It means you can do the same action(like pressing a button) on different things(like different shapes) and get the right response. It makes code flexible and easy to use.
Operator Overloading	Operator overloading is like using a symbol(like a plus sign) in different ways based on what you're working with. It lets you make symbols do special things for

	your stuff. For example, the “+” sign can join words or add numbers, depending on the situation. It’s like giving familiar tools new jobs.