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 (OOP) is a programming is based on the idea of objects that contain data and code. Data is in the form of properties or attributes and code is in the form of procedures. What are objects and classes in Python? Come up with a real-world example to illustrate how objects and classes work.

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

Basically objects are contained within a class. An object can be any data type such as numbers, text, sequences and dictionaries. Classes can contain multiple objects. A real-life example of class-object situation is if a grocery cart is like a class, then attributes like number of groceries, cart number and total cost of groceries can be the objects.

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 in Python is when one class inherits a method from another class. For example, certain attributes from Class A can be inherited from Class B without explicitly stating it.
Polymorphism	Where a given data attribute or method has the same name across different classes or data types but performs different operations depending on where it was defined.
Operator Overloading	To use operators on a custom class, you need to define your own methods for them. This process of using operators on a custom class by defining your own method for them is known as operator overloading. All it requires is defining a function with a name that Python already reserves for your operator and surrounding it with double underscores.