

Python for Web Developers Learning Journal

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?

OOP is the process of abstracting data and methods into classes. Each object has its own attributes and methods. The benefit of OOP is that it follows a DRY (don't-Repeat-Yourself) method which makes code non-repetitive and efficient.

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

A class is a user-defined blueprint or prototype from which objects are created. It defines properties and behaviors which all objects of that class will have. For example, a class called smartphones would be blueprint for each object including attributes such as the smartphone name, manufacturer, the price, and the amount of gigabytes it has. It could also include methods such as "restart phone" or "adjust settings." Each phone would have different ways to perform these operations but they all share the same methods and 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	Inheritance allows you to define a class that inherits all the methods and properties from another class. The class that is being inherited from is the parent class, and the subclass is the class which all data attributes and procedures are being copied over to. The class that you inherit properties from is passed in the function argument.
Polymorphism	Polymorphism is where a given data attribute or method has the same name across different classes or data types, but can execute different operations depending on where it was defined. An example of this is the len() function where it returns different types of data for tuples, dictionaries, and strings
Operator Overloading	Operator overloading allows you to define custom behaviors for standard operators. For example, the + operator will perform addition on two numbers or concatenate two strings.