

Exercise 1.5: Object-Oriented Programming in Python

Learning Goals

- Apply object-oriented programming concepts to your Recipe app

Reflection Questions

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

Object-oriented programming involves building objects and organizing them into classes with data attributes that use methods to interact with the data. Objects are instances of a class or type associated with them that works as a template to organize their data structure.

OOP uses a process to abstract data and methods into classes to access the data. Classes can help create new objects that can be populated.

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

Everything is an object in Python and they are instances of a class. Classes work to organize the object's data into a template that can be reused. An example could be a music genre playlist.

The class would be GenrePlayList

- Data attributes:
 - Genre name
 - Band Name
 - Album Name
 - Artist Names
- Procedural attributes
 - Methods to add genres, bands, albums, and artists to playlist
 - Methods to remove genres, bands, albums, and artists from playlist
 - Methods to search for genres, bands, albums, and artists
- 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 for the use of properties and methods of a class to be used or inherited from another class. The class being inherited is called the parent class and the recipient of the class is called the subclass. The inheritance works by using the inherited object's class and parameters to determine what properties and attributes are to be inherited. The inheritance can only work one way from the parent to the subclass.

Polymorphism	Polymorphism means many forms, and in the case of OOP, it allows for the reuse of a method name across data classes and types, each of which can perform its own operations. The name can be used to specify data attributes and/or methods that can be reused and implemented differently as needed. Polymorphism can also allow for a method to be called without needing to know its class type.
Operator Overloading	Operator overload is called for when operators are used in custom classes, which do not support operators, so they need to be defined using methods. One would need to define an operator function with a name, not an operator sign, (i.e. + vs. add) and it would need to be surrounded by underscores such <code>__add__()</code> to have it work as expected in the custom class.