EXERCISE 1.5: OBJECT-ORIENTED PROGRAMMING IN PYTHON

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

OOP is a process that is based on the concept of building "objects", they can contain data and code that manipulates that data. The objects are instances of classes, which are templates or blueprints for creating objects. One of the most important benefits of using OOP is reusability: OOP allows for the reuse of code using classes and objects, making development faster and more efficient.

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

An object consists of data types that can be used in methods, and classes act as a template that consists blueprints for creating multiple objects and methods. An example of objects would be multiple food items on a menu for a restaurant, while an example of classes would be a collection of attributes food items would consist of and methods for preparing the food items.

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

Method	Description
Inheritance	This is the mechanism of deriving a new class from an existing one. The child class inherits the properties and methods of the parent class.
Polymorphism	This is the ability of an object to take on many forms. The most common use of polymorphism in OOP occurs when a parent class reference is used to refer to a child class object.
Operator Overloading	This is a feature that allows operators (such as +, -, *, etc.) to have different meanings depending on the context in which they are used. This is achieved by creating special methods in a class that are called when an operator is used with an instance of that class.