Day 11:

**Classes**:

* A **class** is a user-defined blueprint or template for creating objects.
* It encapsulates data (attributes) and behavior (methods) related to a specific concept or entity.
* You define a class using the **class** keyword, followed by the class name and a colon. Inside the class, you can define attributes (instance variables) and methods (functions).

Example:

class Person:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

def greet(self):

return f"Hello, my name is {self.name} and I'm {self.age} years old."

# Creating an object of the Person class

person1 = Person("Alice", 30)

print(person1.greet()) # Output: "Hello, my name is Alice and I'm 30 years old."

**Objects**:

* An **object** is an instance of a class.
* When you create an object, memory is allocated for it, and it becomes a concrete representation of the class.
* Objects can access the attributes and methods defined in their class.
* You can create multiple objects from the same class, each with its own unique data.

Example:

person2 = Person("Bob", 25)

print(person2.greet()) # Output: "Hello, my name is Bob and I'm 25 years old."

**Attributes and Methods**:

* + Attributes (also called instance variables) store data specific to an object.
  + Methods are functions defined within a class and operate on the object’s data.
  + You can access attributes using dot notation (e.g., person1.name) and call methods using parentheses (e.g., person1.greet()).