**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Topic 53 - Classes**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**What**

* **Classes** are **templates** in Python, providing a structure for creating objects with a **standardized set of data** and behaviors.
* Using classes helps you **organize code**, manage **related data**, and **reuse structures**.

**Why**

* **Consistency**: Classes enforce a **consistent format** for each object created from them.
* **Reusability**: By defining data and behaviors once in a class, they can be reused as **individual instances** (objects).
* **Easier Maintenance**: Classes make it simpler to **update** or **extend** the program, as you only modify the class definition.

**How**

1. **Creating a Class**  
   This example creates a Patient class that will serve as a template for storing information about clinic patients.

python

Copy code

class Patient:

# Initialization method to define initial attributes

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

self.name = name

self.age = age

self.medical\_condition = medical\_condition

# Method to display patient's information

def display\_info(self):

print(f"Patient Name: {self.name}")

print(f"Age: {self.age}")

print(f"Medical Condition: {self.medical\_condition}")

* + **Class Definition**: The class is named Patient and begins with the keyword class, followed by the class name (capitalized by convention), parentheses, and a colon.
  + **Constructor (\_\_init\_\_)**: Inside the class, we define an \_\_init\_\_ method (also called a constructor) to initialize attributes like name, age, and medical\_condition.
  + **Instance Methods**: The display\_info method is defined to show patient details, using the self keyword to refer to instance-specific data.

1. **Creating an Instance of the Class**

python

Copy code

# Creating an instance of Patient

patient1 = Patient("Alice Smith", 29, "Hypertension")

# Using a method on the instance

patient1.display\_info()

* + **Instantiation**: patient1 is an instance of the Patient class, and it is created by calling Patient("Alice Smith", 29, "Hypertension").
  + **Accessing Methods**: patient1.display\_info() calls the display\_info method, displaying the patient’s data.

**Things to Remember**

* **Classes are capitalized by convention**: Use names like Patient, Employee, etc., to easily identify them.
* **\_\_init\_\_ Method**: This special method is used for **initializing objects** with specific attributes.
* **Use self**: This keyword lets each instance of the class **access its own attributes and methods**.