Topic 57 - Classes: Adding Complexity with Multiple Attributes

**What**

* In this section, we add more attributes to the Patient class, making each instance more detailed. Now, each Patient instance includes last\_name, first\_name, and age, giving more context about each individual.
* Multiple attributes allow instances to store a range of information, making the data structure more representative of real-world use cases.

**Why**

* **Realism and Detail**: Adding attributes such as first\_name and age makes the Patient class more representative of actual patient records.
* **Efficient Data Storage**: Storing multiple pieces of related information in each instance improves organization and enables easy retrieval of complete details for each patient.
* **Order-Based Consistency**: Using multiple attributes demonstrates the importance of argument order when initializing instances, ensuring correct data matching.

**How**

1. **Updating the Class with Multiple Attributes**  
   The class is modified to include additional attributes for first\_name and age.

python

Copy code

class Patient:

def \_\_init\_\_(self, last\_name, first\_name, age):

self.last\_name = last\_name

self.first\_name = first\_name

self.age = age

* + **Attribute Addition**: Now, the Patient class has three attributes, making it more descriptive and functional.
  + **Order Matters**: The order of attributes in the \_\_init\_\_ method (last\_name, first\_name, age) dictates how values are assigned during instantiation.

1. **Creating Instances with Multiple Attributes**  
   Instances now include values for each of the three attributes.

python

Copy code

pid4343 = Patient("Taleb", "Sue", 61)

pid4344 = Patient("Anand", "Punya", 29)

pid4345 = Patient("Oppenheimer", "Douglas", 15)

pid4346 = Patient("Lin", "Lilly", 48)

pid12902 = Patient("Nilsson", "Rhonda", 33)

* + **Positional Matching**: Each value matches the corresponding attribute in order: last\_name, first\_name, and age.
  + **Unique Identifiers**: Instances (like pid4343) remain uniquely identified, and each one holds complete patient data.

**Things to Remember**

* **Positional Arguments**: When creating an instance, the order of values matters, aligning with the order of attributes in the \_\_init\_\_ method.
* **Multiple Attributes**: Adding more attributes makes instances richer and more informative, better suited for real-world data needs.
* **Structure Consistency**: Every instance follows the same structure, which simplifies data access and manipulation within the program.