DATA SCIENCE COURSE TUTORIAL # 35

3.22 Instance Variables and Class Variables

Introduction to Variables in OOP

In Object Oriented Programming, a class can contain two main types of variables. These are instance variables and class variables. Both are used to store data, but they behave differently and are used for different purposes.

What Are Instance Variables

Instance variables are variables that are unique for every object created from a class. Each object gets its own separate copy of these variables. Instance variables are usually defined inside the **init** function using self.

Example:

```
class Student:
    def __init__(self, name, age):
        self.name = name
        self.age = age
```

In this example, name and age are instance variables. Each object of Student will have its own name and age.

Example of Creating Objects:

```
s1 = Student("Ali", 20)
s2 = Student("Aisha", 22)

print(s1.name, s1.age)
print(s2.name, s2.age)
```

Output:

```
Ali 20
Aisha 22
```

What Are Class Variables

Class variables are shared by all objects of the class. They are defined inside the class but outside the **init** function. All objects access the same value of a class variable.

Example:

```
class Student:
    school = "UST Bannu"  # class variable

    def __init__(self, name, age):
        self.name = name     # instance variable
        self.age = age       # instance variable
```

Here, school is a class variable. It will be same for every Student object unless changed at class level.

Example of Using Class Variables:

```
s1 = Student("Ali", 20)
s2 = Student("Aisha", 22)

print(s1.school)
print(s2.school)
```

Output:

```
UST Bannu
UST Bannu
```

Changing Class Variables

If you change a class variable using the class name, the change will be reflected for all objects.

Example:

```
Student.school = "ABC University"

print(s1.school)
print(s2.school)
```

Output:

```
ABC University
ABC University
```

If you change a class variable through an object, Python creates a new instance variable for that object only.

Example:

```
s1.school = "XYZ College"

print(s1.school)
print(s2.school)
```

Output:

```
XYZ College
ABC University
```

This happens because s1 now has its own instance variable named school.

Difference Between Instance Variables and Class Variables

Instance variables store data that is different for each object. Class variables store data that is common for all objects.

Comparison Example:

```
class Car:
    wheels = 4  # class variable

    def __init__(self, color, model):
        self.color = color       # instance variable
        self.model = model       # instance variable

c1 = Car("Red", "2024")
c2 = Car("Blue", "2023")

print(c1.color, c1.model, c1.wheels)
print(c2.color, c2.model, c2.wheels)
```

Output:

```
Red 2024 4
Blue 2023 4
```

Why Use Instance and Class Variables

Instance variables are useful when each object needs its own data. Class variables are useful when the same value should be shared across all objects. Together they help build flexible and powerful class structures.

Summary

- Instance variables belong to individual objects.
- Class variables are shared among all objects of a class.
- Instance variables are defined inside the **init** function.
- Class variables are defined inside the class but outside any method.
- Changing class variables affects all objects unless overridden by an instance.

Common Use Cases

- Instance variables are used to store personal data like name, age, color or size.
- Class variables are used for constants like school name, company name or number of wheels.