Day 34

Constructors in Python:

What is a Constructor?

A constructor is a method named **__init__()** (double underscore before and after init).

Syntax

class ClassName:
def __init__(self):
 # constructor body
pass

- __init__ runs automatically when an object is created.
- self refers to the current object.

Example: till now we know.

```python
class Person:
    name = "Aditya"
    occ = "SDE"
    #method
    def info(self):
        print(f"{self.name} is of {self.occ}")
#creating the object
a = Person()
print(a.name)
a.info()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\Python\Day31-40\Day34> python main.py
Aditya
Aditya is of SDE
```

Example: creating a constructor.

```python
class Person:
    def __init__(self):
        print("Hey I am a person")
    #method
    def info(self):
        print(f"{self.name} is of {self.occ}")
#creating the object
a = Person()
```

```
PS E:\Python\Day31-40\Day34> python main.py
Hey I am a person
```

Example: whenever a new object is created, the constructor is called. Since, two objects are there, two time “Hey I am a person” is printed.

```python
class Person:
    def __init__(self):
        print("Hey I am a person")
    #method
    def info(self):
        print(f"{self.name} is of {self.occ}")
#creating the object
a = Person()
b = Person() #2nd object
```

```
PS E:\Python\Day31-40\Day34> python main.py
Hey I am a person
Hey I am a person
```

Example: passing multiple arguments in the constructor.

```python
class Person:
    def __init__(self, n, o):
        print("Hey I am a person")
        self.name = n
        self.occ = o
    #method
    def info(self):
        print(f"{self.name} is of {self.occ}")
#creating the object
a = Person("Aditya", "SDE")
b = Person("Utsav", "BHMS") #2nd object
a.info()
b.info()
```

```
PS E:\Python\Day31-40\Day34> python main.py
Hey I am a person
Hey I am a person
Aditya is of SDE
Utsav is of BHMS
```

Example: passing only one argument instead of two.

```python
class Person:
    #constructor
    def __init__(self, n, o):
        print("Hey I am a person")
        self.name = n
        self.occ = o
    #method
    def info(self):
        print(f"{self.name} is of {self.occ}")
#creating the object
a = Person("Aditya")
b = Person("Utsav", "BHMS") #2nd object
a.info()
b.info()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\Python\Day31-40\Day34> python main.py
Traceback (most recent call last):
  File "E:\Python\Day31-40\Day34\main.py", line 11, in <module>
    a = Person("Aditya")
TypeError: Person.__init__() missing 1 required positional argument: 'o'
```

Example: default constructor.

```python
class Demo:
    def __init__(self):
        self.x = 10
d = Demo()
print(d.x)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POI

```
PS E:\Python\Day31-40\Day34> python main.py
10
```

Example: parameterizes constructor.

```python
class Demo:
    def __init__(self, x, y):
        self.x = x
        self.y = y
d = Demo(5, 10)
print(d.x, d.y)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POR

```
PS E:\Python\Day31-40\Day34> python main.py
5 10
```

Summary:

- A constructor is a special method that is automatically executed when an object is created.
- In Python, a constructor is defined using __init__().
- The main purpose of a constructor is to initialize instance variables.
- self represents the current object of the class.
- Constructors are not mandatory, but commonly used.

Key Points

- Constructor name must be __init__ (with double underscores).
- It runs automatically when an object is created.
- Used to assign initial values to object data members.
- A class can have only one constructor (method overriding applies).
- Python does not support constructor overloading directly.

Types of Constructors

- Default Constructor
    - Has no parameters (except self)
- Parameterized Constructor
    - Accepts parameters to initialize data members

Special Notes

- Constructor overloading can be simulated using default arguments.
- Constructors improve code readability and structure.
- If no constructor is defined, Python provides a default constructor.

--The End--