Chapter 1 - Class

Theory:

Classes are blueprints for creating objects.

Code Example:

```
# ■ 1. What is a Class?
# A class is a blueprint for creating objects (instances).
# It groups data (attributes) and behavior (methods) together.
# ■ Real-Life Analogy:
# Think of a Car class as a blueprint.
# Attributes: color, model, speed
# Methods: start(), stop(), accelerate()
# Each real car (like Maruti, BMW) is an object created from that blueprint.
# ■ 2. Basic Class Syntax
class Car:
def __init__(self, brand, color):
self.brand = brand # instance attribute
self.color = color
def start(self): # instance method
print(f"{self.brand} is starting!")
# Creating an object of Car
my_car = Car("Toyota", "Red")
# Accessing attributes and methods
print(my_car.color)
my_car.start()
# ■ 3. Attributes and Methods
# ➤ Instance Variables (unique to each object)
# Defined using self inside __init__ or methods
# Example: self.brand = brand
# ➤ Instance Method
# Defined using self parameter
# Example: def start(self): ...
# ➤ Class Variables (shared by all instances of the class)
class Car:
wheels = 4 # class variable (shared)
# ■ Class with class and instance attributes
class Employee:
# Class attributes (shared among all instances)
name = "harry"
lang = "hindi" # this is a class attribute
salary = 12000 # class attribute
# Creating object 'harry'
harry = Employee() # object created
harry.name = 'harry' # instance attribute (overrides class attribute)
print(harry.lang, harry.salary) # accessing class attributes
# Creating another object 'rohan'
rohan = Employee()
rohan.name = 'rohan' # instance attribute
print(rohan.lang, rohan.salary) # accessing class attributes
# ■ Notes:
# - 'name' is an object (instance) attribute because we manually assigned it to each
object.
# - 'lang' and 'salary' are class attributes — shared across all objects unless overridden.
```

```
class Student:
def __init__(self, name, age):
self.name = name
self.age = age
def show(self):
print("Name:", self.name)
print("Age:", self.age)
# Object create karna
s1 = Student("Milan", 22)
# Method call karna
s1.show()
class Calculator:
def add(self, x, y):
print("Sum is:", x + y)
c = Calculator()
c.add(5, 3)
# ■ Class = Design
# ■ Object = Us design ka actual item
# ■ __init__() = Constructor
# ■ self = Har object ke liye alag-alag data
```