Inheritance

nheritance is an OOP concept that allows a class (called child or subclass) to inherit properties and methods from another class (called parent or superclass).

This promotes code reusability, reduces redundancy, and helps in creating a hierarchical structure.

Types of Inheritance in Python:

1. Single Inheritance – A child inherits from one parent class.
2. Multiple Inheritance – A child inherits from more than one parent class.
3. Multilevel Inheritance – Inheritance chain, e.g., A → B → C.
4. Hierarchical Inheritance – Multiple children inherit from one parent.
5. Hybrid Inheritance – Combination of two or more types above.

Single Inheritance: A child class inherits from one parent class.

```
# Parent Class
class Vehicle:
    def __init__(self, brand):
        self.brand = brand

    def show_brand(self):
        print(f"Brand: {self.brand}")

# Child Class
class Car(Vehicle):
    def __init__(self, brand, model):
        super().__init__(brand)
        self.model = model

    def show_model(self):
        print(f"Model: {self.model}")

my_car = Car("Tata", "Nexon")
my_car.show_brand()   # Inherited from Vehicle
my_car.show_model()   # Defined in Car

# Output
Brand: Tata
Model: Nexon
```

Multiple Inheritance: A child class inherits from more than one parent class.

```
class Engine:
    def start_engine(self):
        print("Engine started")

class Vehicle:
    def move(self):
        print("Vehicle is moving")

class Car(Vehicle, Engine):
    pass

my_car = Car()
my_car.start_engine()  # From Engine
my_car.move()          # From Vehicle

# Output
Engine started
Vehicle is moving
```

Multilevel Inheritance: Inheritance chain (Grandparent → Parent → Child).

```
class Vehicle:
    def info(self):
        print("This is a vehicle")

class Car(Vehicle):
    def type(self):
        print("This is a car")

class ElectricCar(Car):
    def battery(self):
        print("Electric car battery info")

ecar = ElectricCar()
ecar.info()      # From Vehicle
ecar.type()      # From Car
ecar.battery()   # From ElectricCar

# Output
This is a vehicle
This is a car
Electric car battery info
```

Hierarchical Inheritance: Multiple children inherit from the same parent class.

```
class Vehicle:
    def fuel(self):
        print("Vehicle fuel info")

class Car(Vehicle):
    def car_type(self):
        print("Car type: Sedan")

class Bike(Vehicle):
    def bike_type(self):
        print("Bike type: Sports")

car = Car()
bike = Bike()

car.fuel()        # Inherited from Vehicle
car.car_type()
bike.fuel()       # Inherited from Vehicle
bike.bike_type()

# Output
Vehicle fuel info
Car type: Sedan
Vehicle fuel info
Bike type: Sports
```

Hybrid Inheritance: Combination of two or more types of inheritance.

```
# Parent class
class Vehicle:
    def info(self):
        print("Vehicle info")

# Child class 1
class Car(Vehicle):
    def car_model(self):
        print("Car model info")

# Child class 2
class Bike(Vehicle):
    def bike_model(self):
        print("Bike model info")
```

```python
# Sub-child inherits from Car and Bike (Multiple + Multilevel)
class ElectricCar(Car, Bike):
    def battery(self):
        print("Electric car battery info")

ecar = ElectricCar()
ecar.info()       # From Vehicle
ecar.car_model()  # From Car
ecar.battery()    # Own method

# Output
Vehicle info
Car model info
Electric car battery info
```