Inheritance

A class can inherit properties and methods from another class.

Basic Inheritance: One class inherits from another.

```
# Parent class
class Animal:
    def speak(self):
        print("Animals make sounds")
# Child class
class Dog(Animal):
    def bark(self):
        print("Dog barks”)

d = Dog()
d.speak()   # Inherited method
d.bark()    # Own method
```

Multilevel Inheritance: A class inherits from a child class (grandchild structure).

```
class Animal:
    def eat(self):
        print("Animal eats")

class Mammal(Animal):
    def walk(self):
        print("Mammal walks")

class Dog(Mammal):
    def bark(self):
        print("Dog barks")

d = Dog()
d.eat()    # From Animal
d.walk()   # From Mammal
d.bark()   # From Dog
```

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

```
class Father:
    def f_skills(self):
        print("Father: Gardening and Programming")

class Mother:
    def m_skills(self):
        print("Mother: Cooking and Art")

class Child(Father, Mother):
    def skills(self):
        # Optionally call parent methods
        Father.f_skills(self)
        Mother.m_skills(self)
        print("Child: Music and Dancing")
        print()

c = Child()
c.skills()
c.f_skills()
c.m_skills()
```

Hierarchical Inheritance: Multiple child classes inherit from the same parent.

```
class Animal:
    def speak(self):
        print("Animal speaks")

class Dog(Animal):
    def bark(self):
        print("Dog barks")

class Cat(Animal):
    def meow(self):
        print("Cat meows")
```

```
d = Dog()
c = Cat()

d.speak()
d.bark()
c.speak()
c.meow()
```

super() Keyword (Calling Parent Methods): Used to call a method from the parent class inside the child class.

```
class Animal:
    def __init__(self):
        print("Animal created")

class Dog(Animal):
    def __init__(self):
        super().__init__()  # Calls Animal constructor
        print("Dog created")
d = Dog()
```

Method Overriding: When a child class defines a method with the same name as in the parent class.

```
class Animal:
    def speak(self):
        print("Animal makes sound")

class Dog(Animal):
    def speak(self):  # Override
        print("Dog barks")

d = Dog()
d.speak()
```