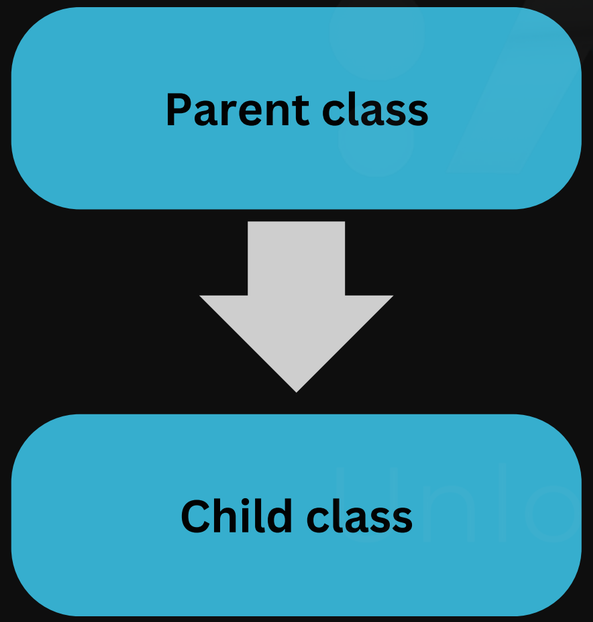
Python Inheritance

Inheritance is one of the fundamental concepts of Object-Oriented Programming (OOP) in Python. It allows a class to inherit attributes and methods from another class, promoting code reusability and hierarchy.

Pankaj Chouhan
www.codeswithpankaj.com

What is Inheritance

Inheritance is a mechanism where one class derives properties and behaviors (methods) from another class.



Base class / Superclass
The class whose properties are inherited.

Derived class / Subclass
The class that inherits from another class

Pankaj Chouhan
www.codeswithpankaj.com

Why Use Inheritance ?

- **Code Reusability :**

  Avoids duplication of code.

- **Improves Maintainability :**

  Changes in the parent class reflect in the child class.

- **Encapsulation :**

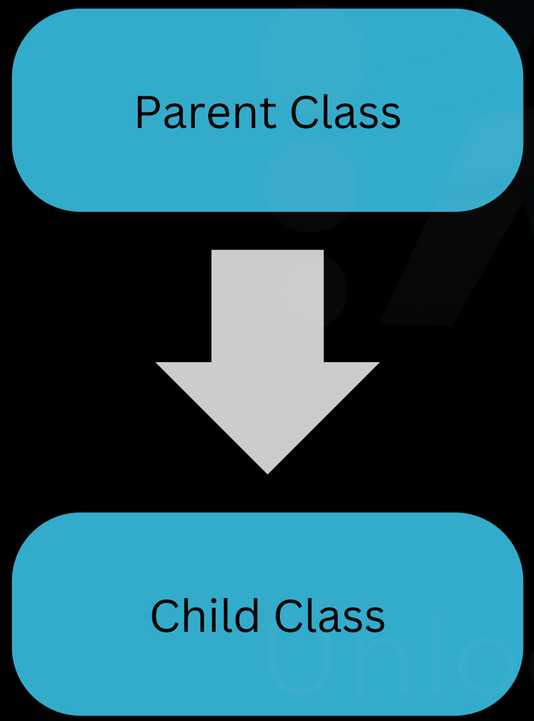
  Allows you to structure your code in a hierarchical way

Types of Inheritance

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

Single Inheritance

A child class inherits from a single parent class.



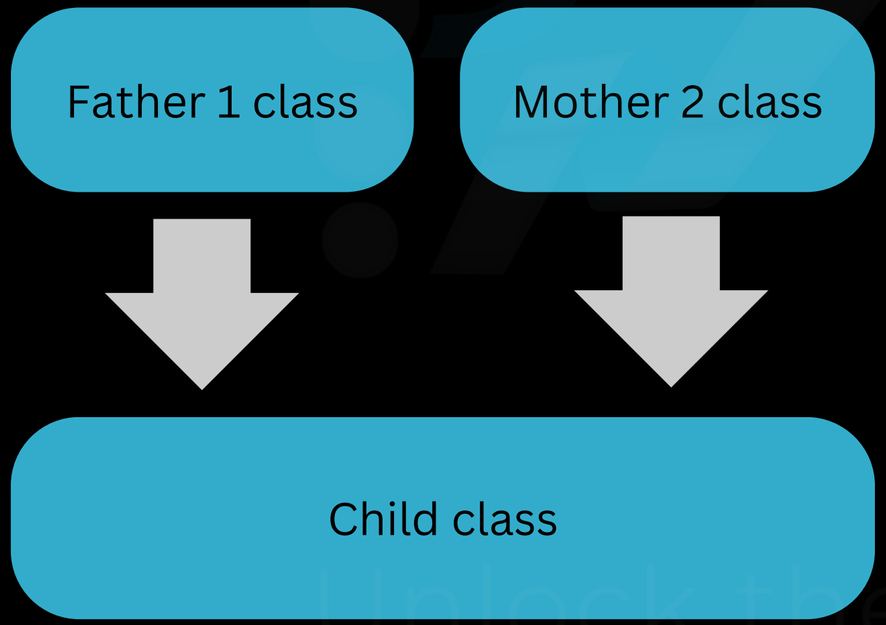
Example

```
class Parent:
    def parent_method(self):
        return "This is parent class"

class Child(Parent):
    def child_method(self):
        return "This is child class"

# Usage
child = Child()
print(child.parent_method())
# Outputs: This is parent class
print(child.child_method())
 # Outputs: This is child class
```


Pankaj Chouhan
www.codeswithpankaj.com

Multiple Inheritance

In multiple inheritance, a child class inherits from more than one parent class.



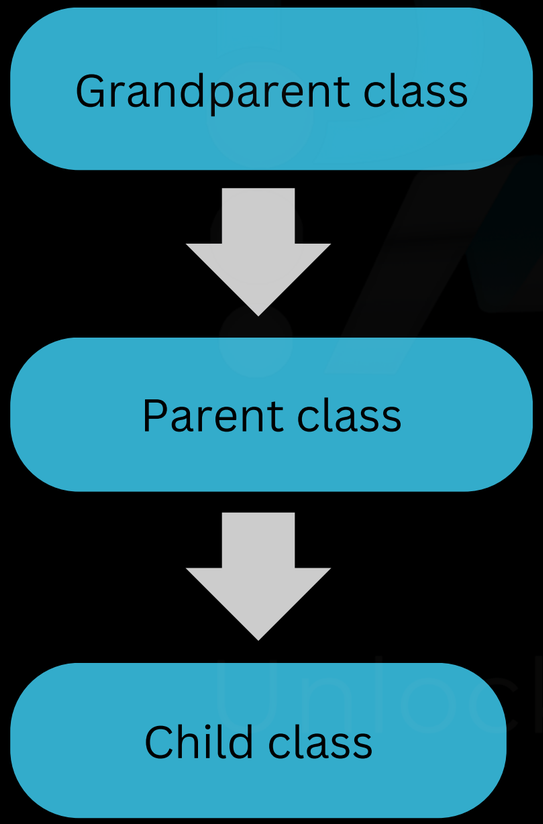
Pankaj Chouhan
www.codeswithpankaj.com

Example

```
class Father:
    def father_method(self):
        return "Father's trait"

class Mother:
    def mother_method(self):
        return "Mother's trait"

class Child(Father, Mother):
    def child_method(self):
        return "Child's trait"

# Usage
child = Child()
print(child.father_method())
# Outputs: Father's trait
print(child.mother_method())
# Outputs: Mother's trait
```

Multilevel Inheritance

In multilevel inheritance, a child class inherits from a parent class, and another child class inherits from that child class.



Pankaj Chouhan

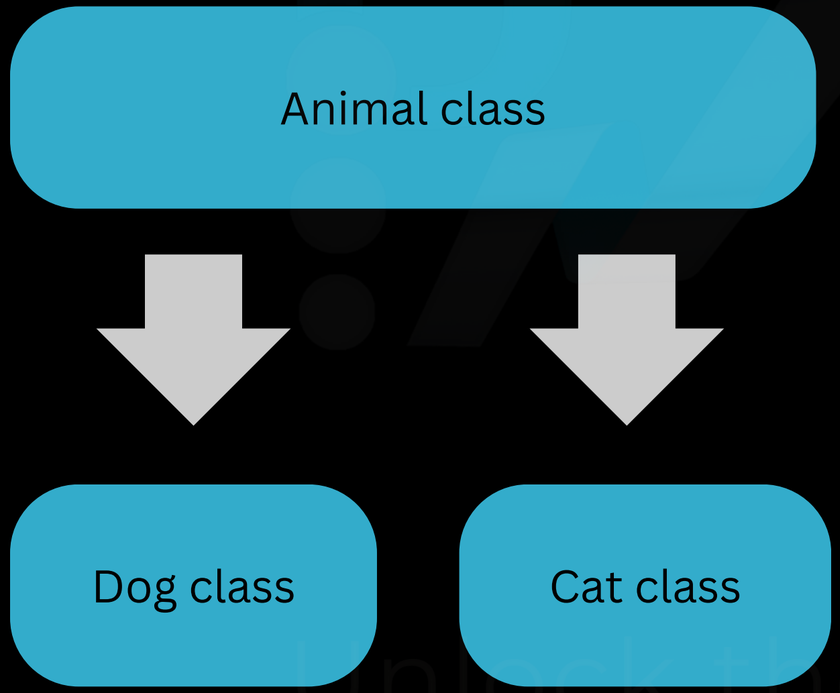
www.codeswithpankaj.com

Example

```
class Grandparent:
    def grandparent_method(self):
        return "Grandparent's method"

class Parent(Grandparent):
    def parent_method(self):
        return "Parent's method"

class Child(Parent):
    def child_method(self):
        return "Child's method"

# Usage
child = Child()
print(child.grandparent_method())
# Outputs: Grandparent's method
print(child.parent_method())
# Outputs: Parent's method
```

Hierarchical Inheritance

In hierarchical inheritance, multiple child classes inherit from a single parent class.



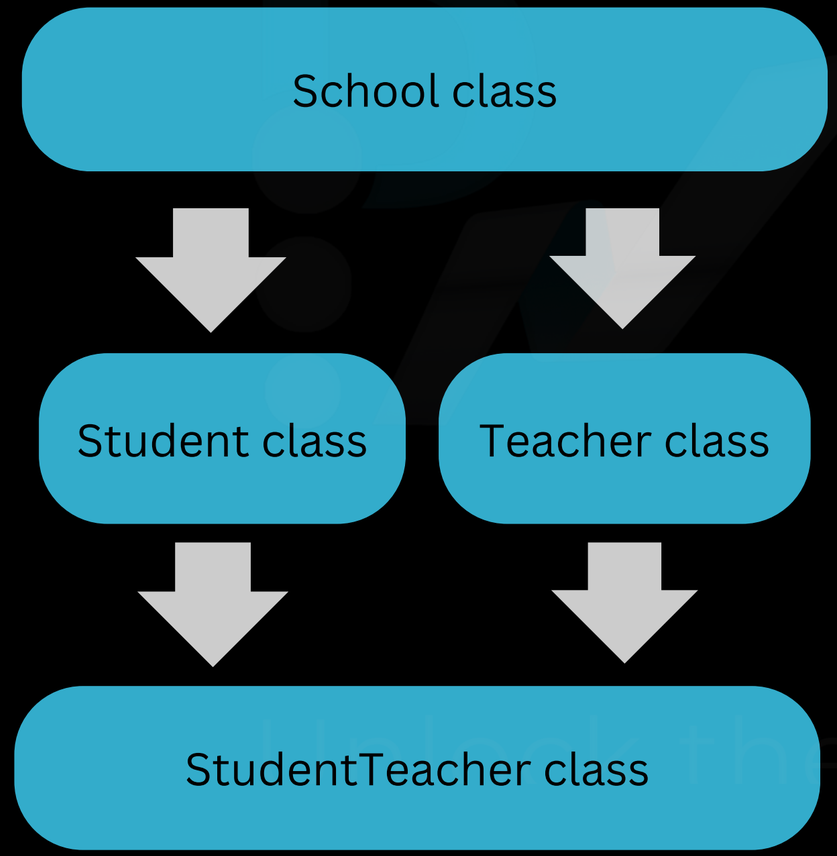
Example

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

class Dog(Animal):
    def speak(self):
        return "Dog barks"

class Cat(Animal):
    def speak(self):
        return "Cat meows"

# Usage
dog = Dog()
cat = Cat()
print(dog.speak())  # Outputs: Dog barks
print(cat.speak())  # Outputs: Cat meows
```

Pankaj Chouhan
www.codeswithpankaj.com

Hybrid Inheritance

Hybrid inheritance is a combination of two or more types of inheritance.



Pankaj Chouhan

www.codeswithpankaj.com

Example

```
class School:
    def school_name(self):
        return "ABC School"

class Student(School):
    def student_info(self):
        return "Student class"

class Teacher(School):
    def teacher_info(self):
        return "Teacher class"

class StudentTeacher(Student, Teacher):
    def student_teacher_info(self):
        return "Student Teacher class"

# Usage
st = StudentTeacher()
print(st.school_name())      # Outputs: ABC School
print(st.student_info())     # Outputs: Student
class
print(st.teacher_info())     # Outputs: Teacher class
```