2. Inheritance:

Inheritance is the process by which one class can takes on the attributes and methods of another class. Newly formed classes are called *child classes* and the classes that child classes are derived from are called *Parent Classes*.

Child classes inherit all of the parent's class attributes and methods, but can also specify different attributes and methods that are unique to themselves or even redefine methods from their parent class.

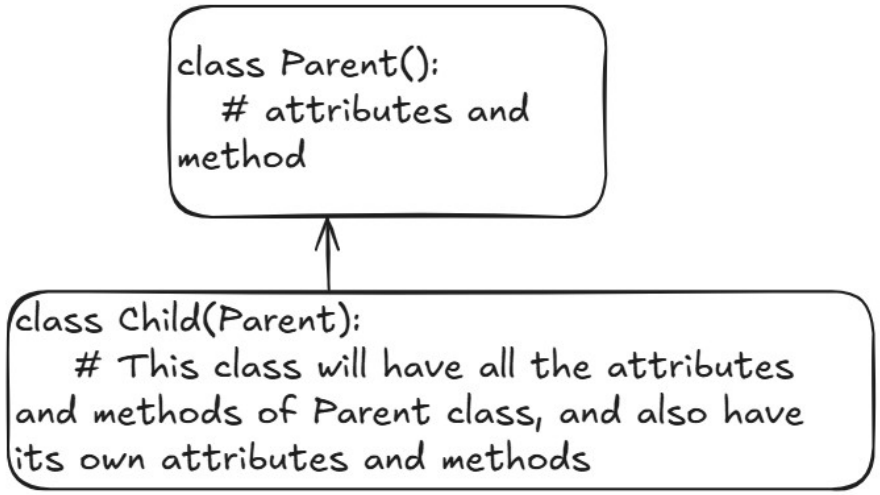
For example:

- You have inherited your hair color from your mother. It's an attribute you were born with. You may decide that you want to color your hair purple. Assuming your mother doesn't have purple hair, you have just **overridden** the hair color attribute.
- You also inherit your language from your parents. If your parents speaks English, then you will also speak English. One day, you may decide to learn a German Language. In this case you are **extending** attributes, because you have added an attribute that your parents do not have.

✓ Single-level Inheritance:

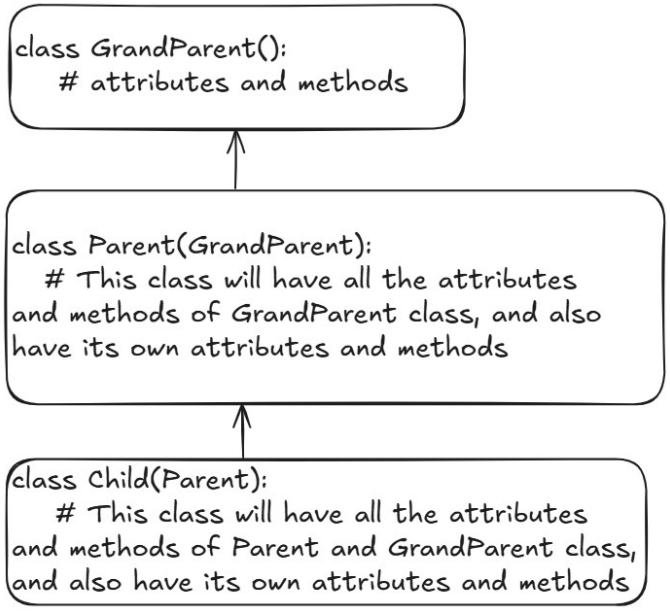
Single level inheritance means **child class inherits from one parent class**.

```
# Parent Class
class Person:
    def __init__(self, name):
        self.name = name

    def display(self):
        print(f"Name: {self.name}")

# Child Class (inherits from Person)
class Student(Person):
    def __init__(self, name, roll_no):
        super().__init__(name)  # Call parent constructor
        self.roll_no = roll_no

    def show(self):
        print(f"Roll No: {self.roll_no}")

# Demo
s1 = Student("Ahsan", 101)
s1.display()   # From Parent
s1.show()      # From Child
```



✓ Multi-level Inheritance:

Multi level inheritance means **a child inherits from a parent class, which inherits itself from another parent class, forming a chain**.



✓ Multiple Inheritance:

A child class inherits from more than one parent class.