

Encapsulation:

- To make protected attributes

↳ for e.g: self.name

↳ self._name

↳ single underscore

for protected

- To make private attributes.

↳ for e.g: self.name

↳ self.__name

↳ private → double underscore

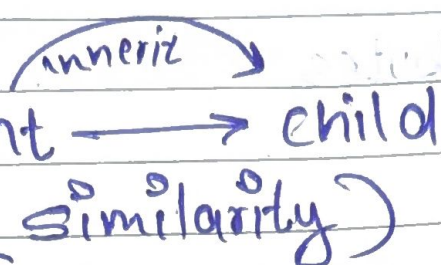
{ for Security &
checks }

- To access these attributes we make getters

& setters.

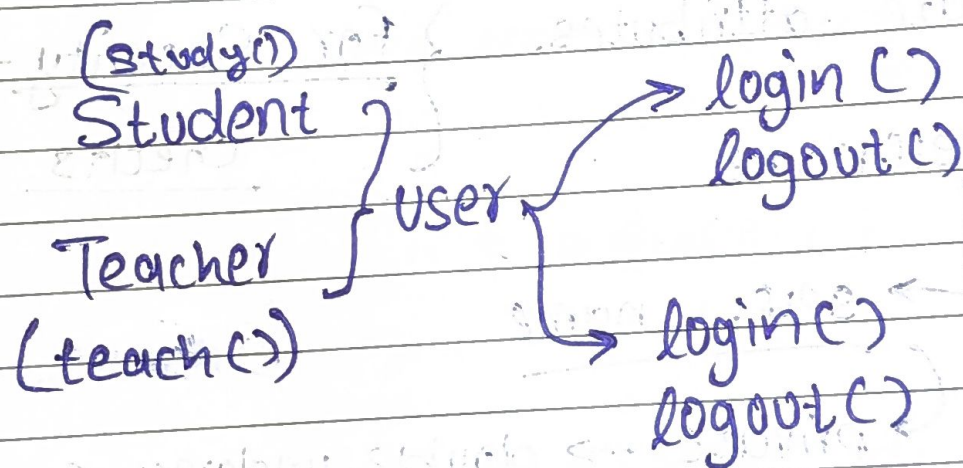
Project: BankAccount

Inheritance :

•  Parent $\xrightarrow{\text{inherit}}$ child
(similarity)

• Gives us major benefit of code reusability

For eg :



• Class User : $\xrightarrow{\text{parent class}}$

```

def login(self):
    return "logged in"
  
```

```

def logout(self):
    return "logged out"
  
```

• Class Student(User): $\xrightarrow{\text{child class}}$

```

pass
  
```


- Things that child class inherits from a parent class:

- Non-private attributes or methods
- constructor and other magic methods

- Parent class can't inherit anything from child.

⇒ Super keyword:

- Super() keyword is used to access methods of parent from our child class.

→ Cannot be used outside.

→ Can only access methods and not attribute.