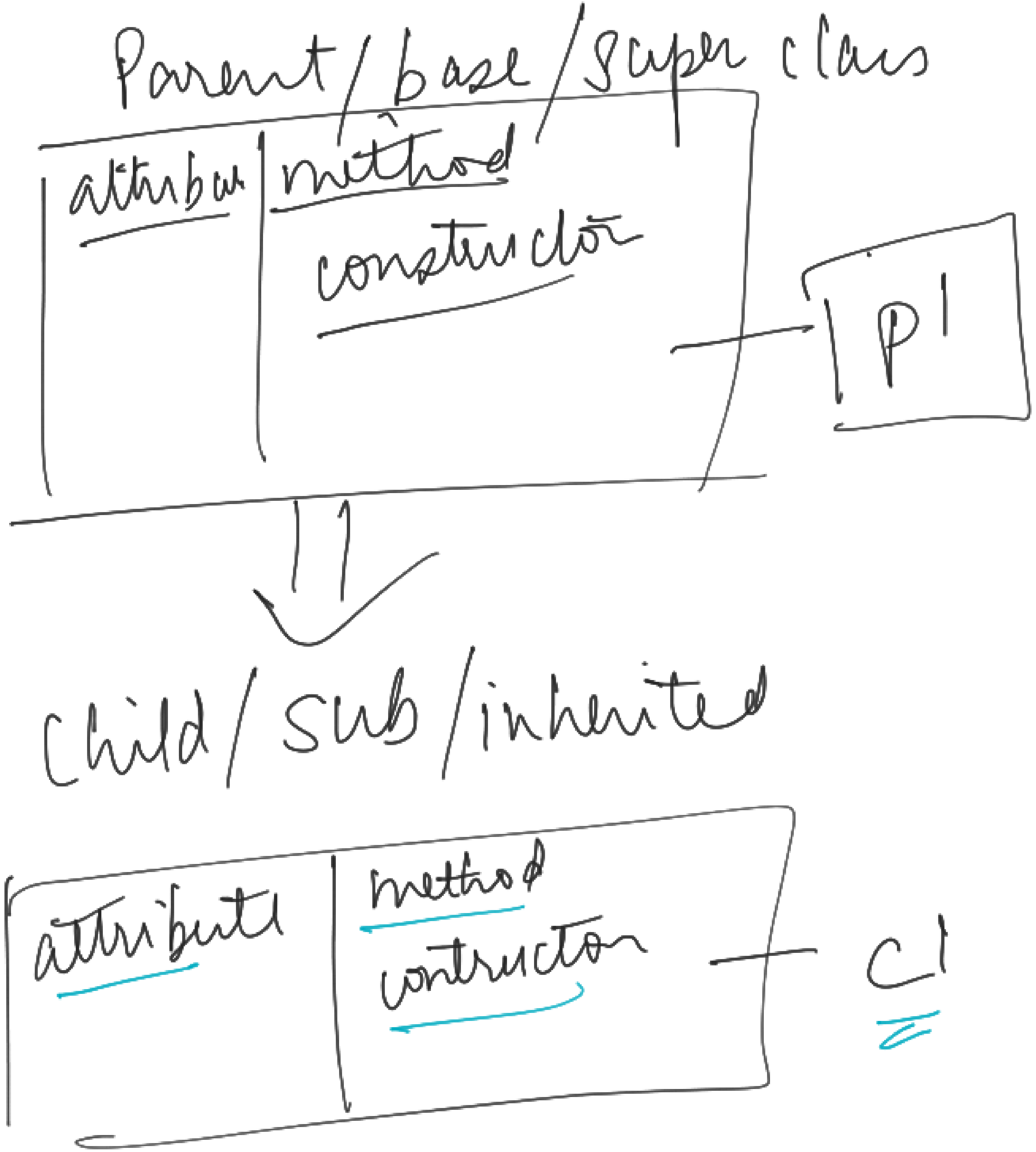
☑ attribute

□ method

□ constructor



Parent → Child(Parent)

count = 0

num = 5

value = 7

num1 = 70

num2 = 80

num3 = 90

p1.count
num
value

c1.num1
.num2
.num3

c1.count
.num
.value

Parent → Child(Parent)

def greet():
    "This is parent class"

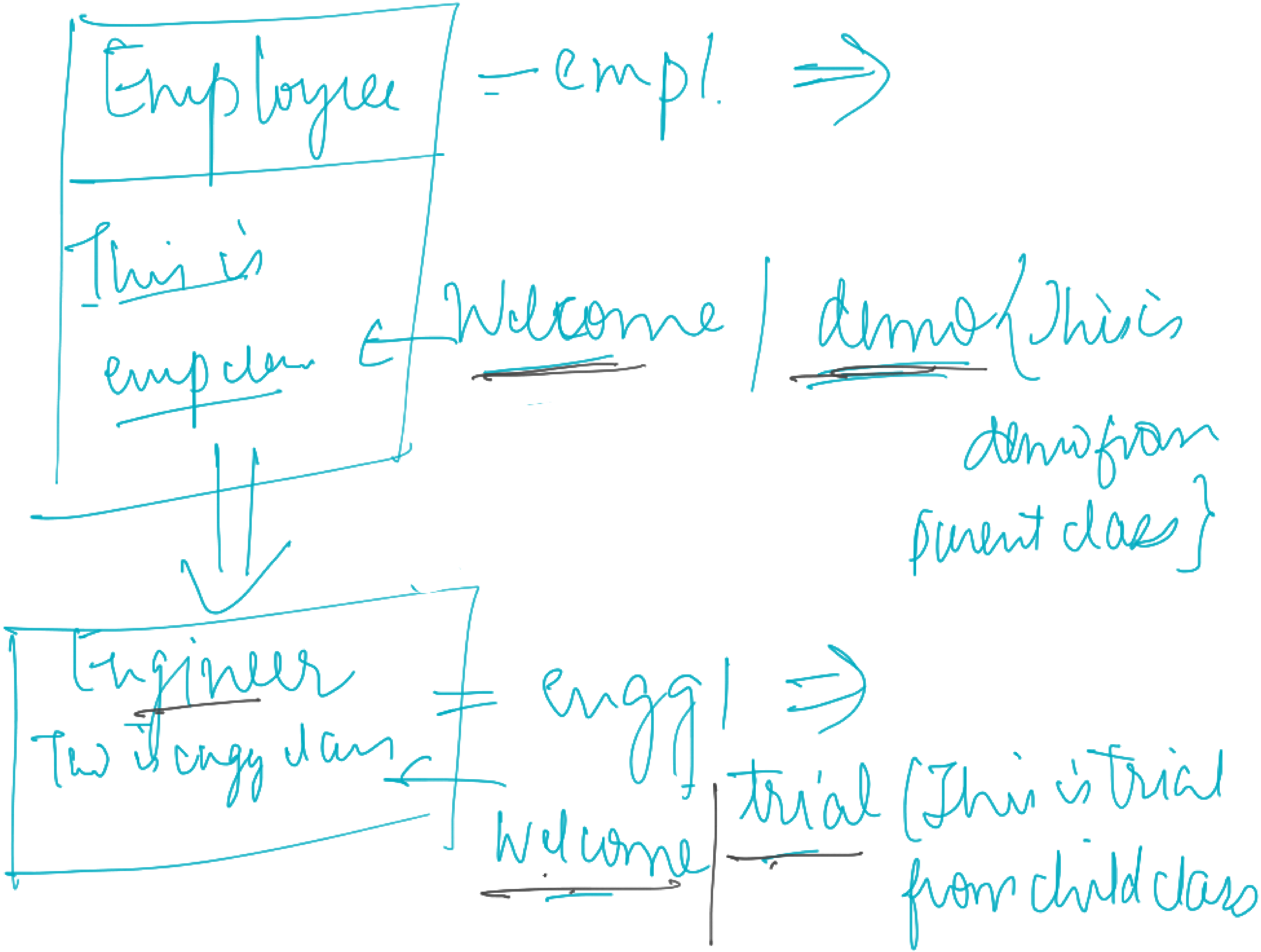
p1.greet

def ~~display():~~ greet()
    "This is child class"

c1.display
c1.greet

Parent

p1. welcome
 . demo


Employee
This is emp clas
= empl. ⇒
Welcome | demo {This is demo from parent class}
Engineer
This is engg clas
= engg1 ⇒
Welcome | trial {This is trial from child class

Child

c1. welcome
demo
trial

Employee ⟹

```
def __init(self, empid):
    self.empid = empid
```

```
emp1 = Employee(1)
```

Engineer

```
~~def __init__(self, branch)~~
~~self.branch = branch~~
```

```
engg1 = Engineer(
```

If both parent & child have different methods:
Child can access methods from both.

If both parent & child have same name methods:
parent and child will call their own method.

If both parent & child have constructor, both will call
their own constructor,

If only the parent has the constructor, child will inherit
from the parent.