03 - 05 - 2025

Agenda :—

Packages

OOP → Class & object → self
__init__
methods
attributes

- Inheritance
- Encapsulation
- polymorphism
- Abstraction

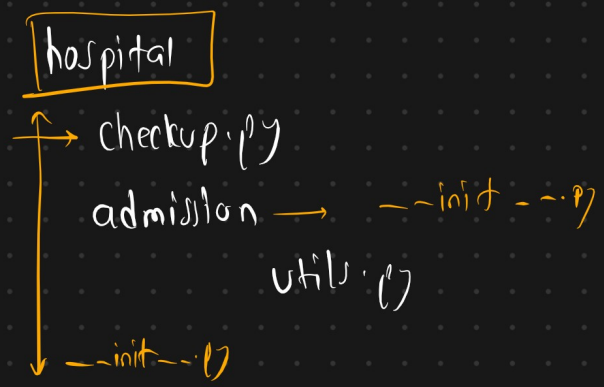
Module → .py file → function → → modular programming
class → → cleaner code
→ easy to maintain

calc.py →

package : → collection of modules

Maths

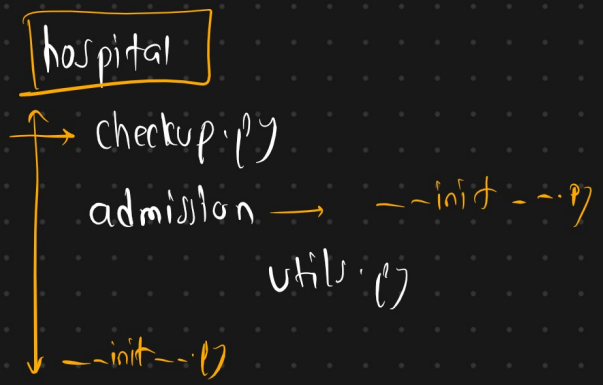
→ algebra.py
trignometry.py
stats.py
calculus.py



hospital. checkup import *

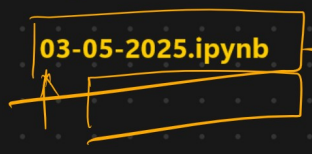
hospital . admission

① create a __init__.py inside a package directory

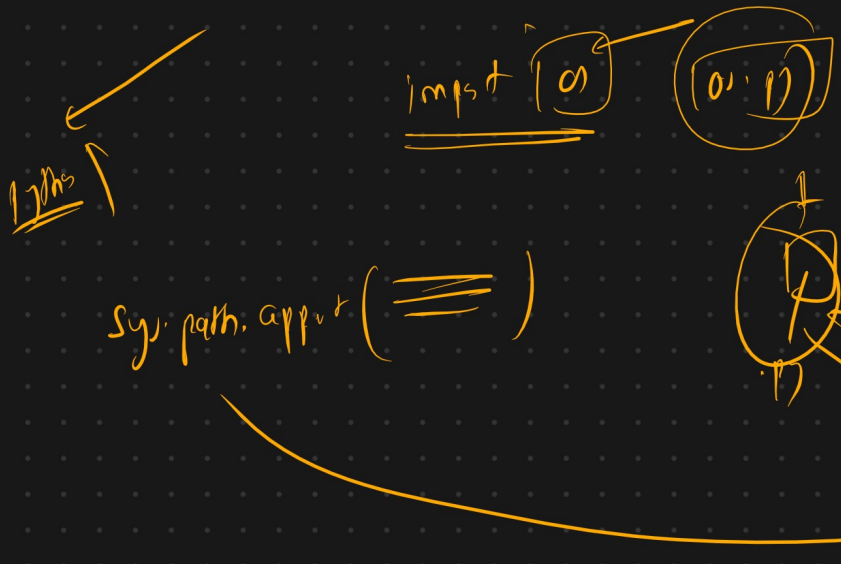


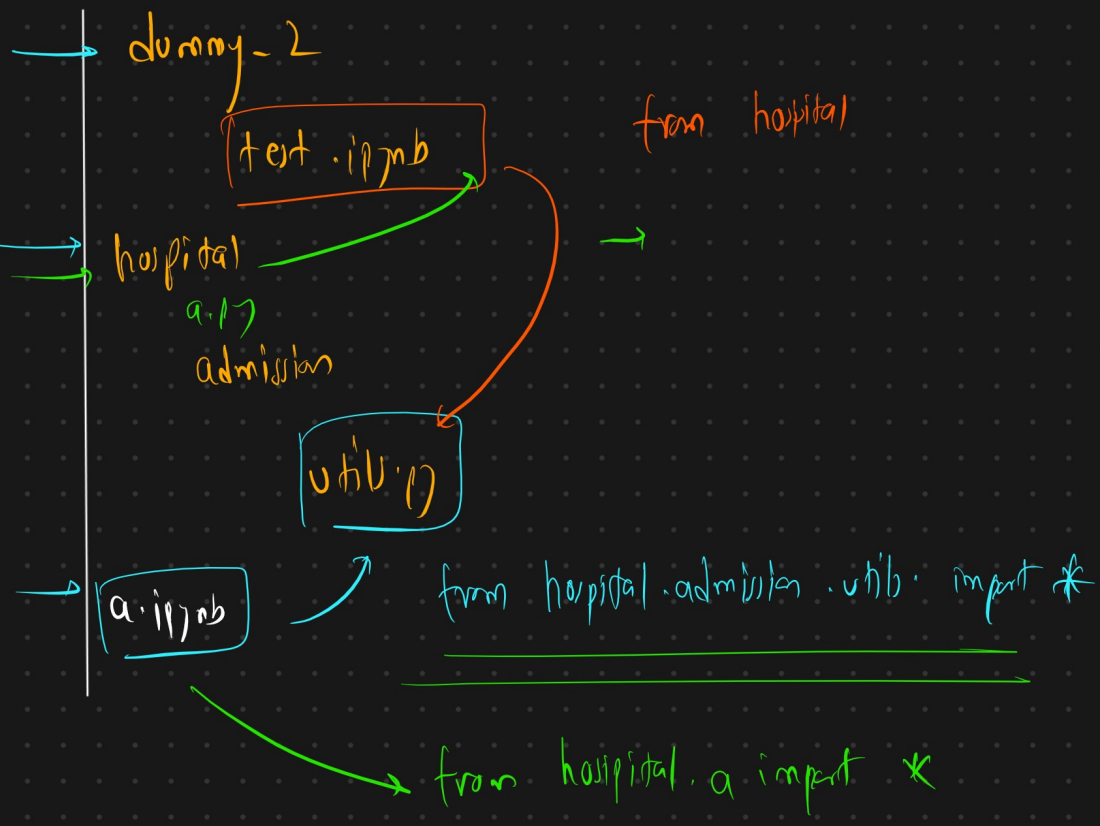
03-05-2025.ipynb

dummy



from hospital



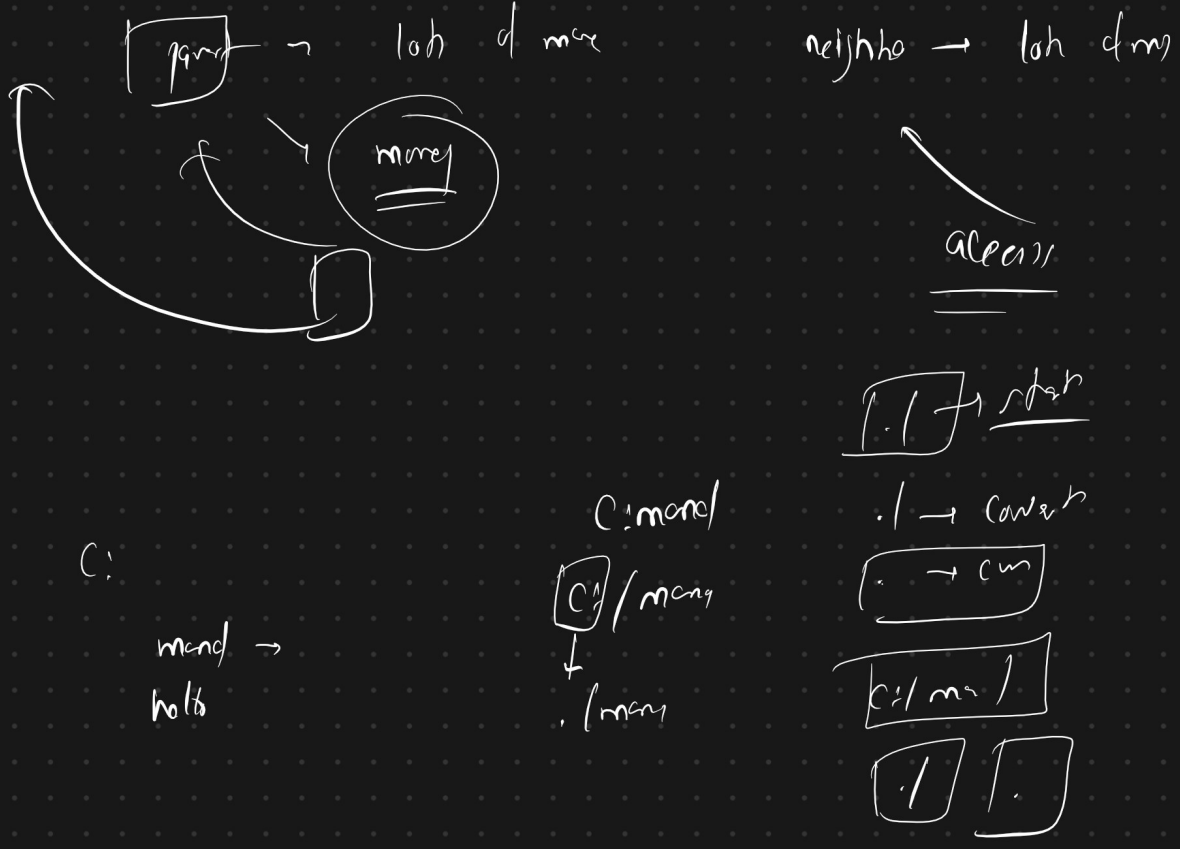


module

package

OOP's

Inheritance → we are inheriting properties from our parent :

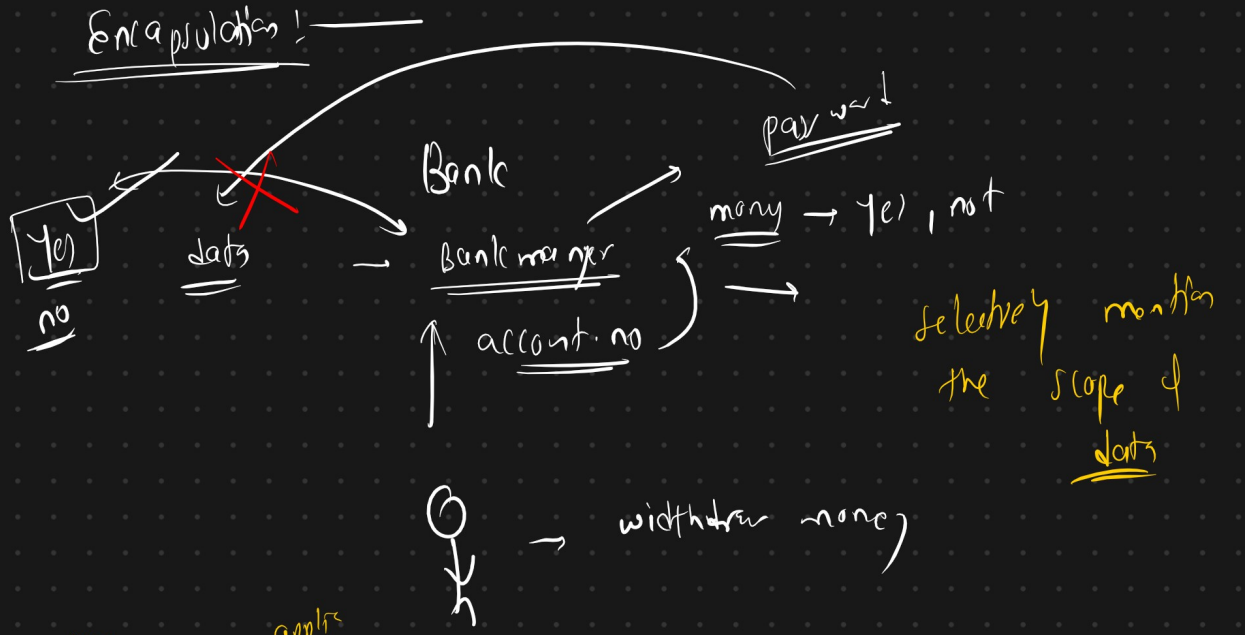


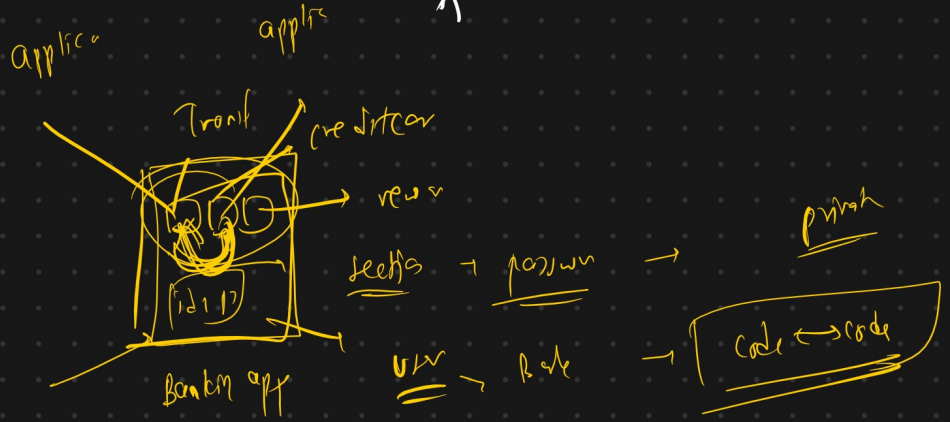

```
class TV:
    def turn_on(self):
        print("TV is now ON.")

class AC:
    def turn_on(self):
        print("AC is cooling the room.")
```

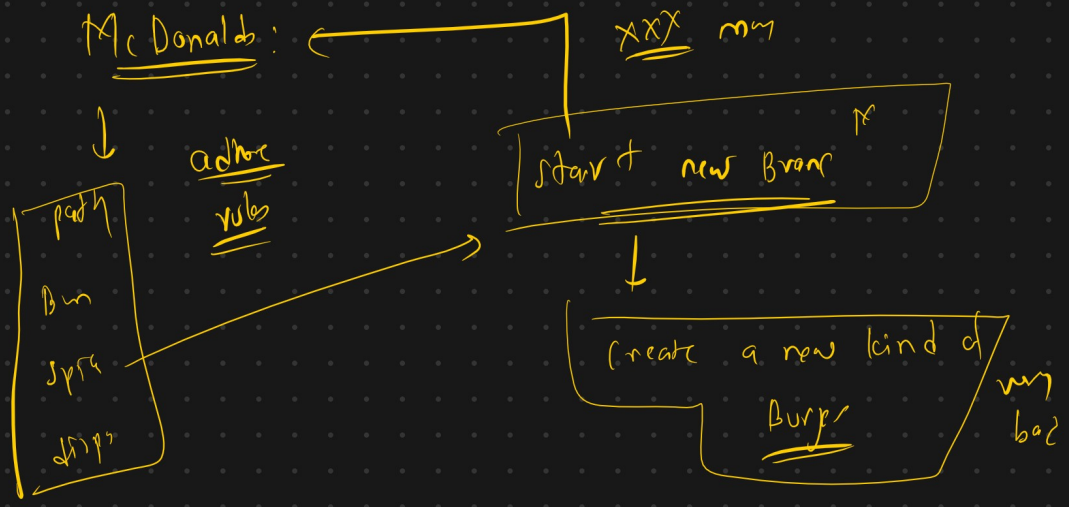
```
def activate(var):
    var.turn_on()
```

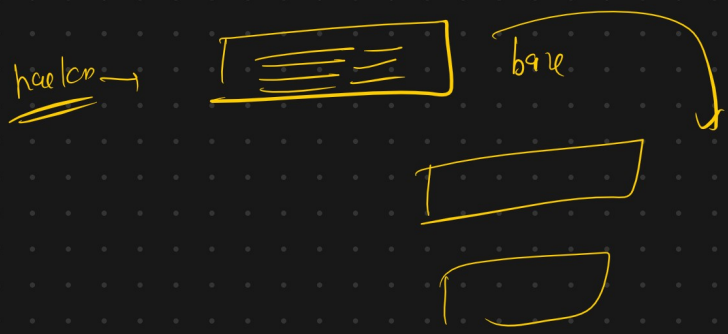
```
class TV1990:
    def turn_on(self):
        print("1990 TV is now ON.")

class TV2000(TV1990):
    def turn_on(self):
        print("2000 TV is now ON.")
```

method overriding







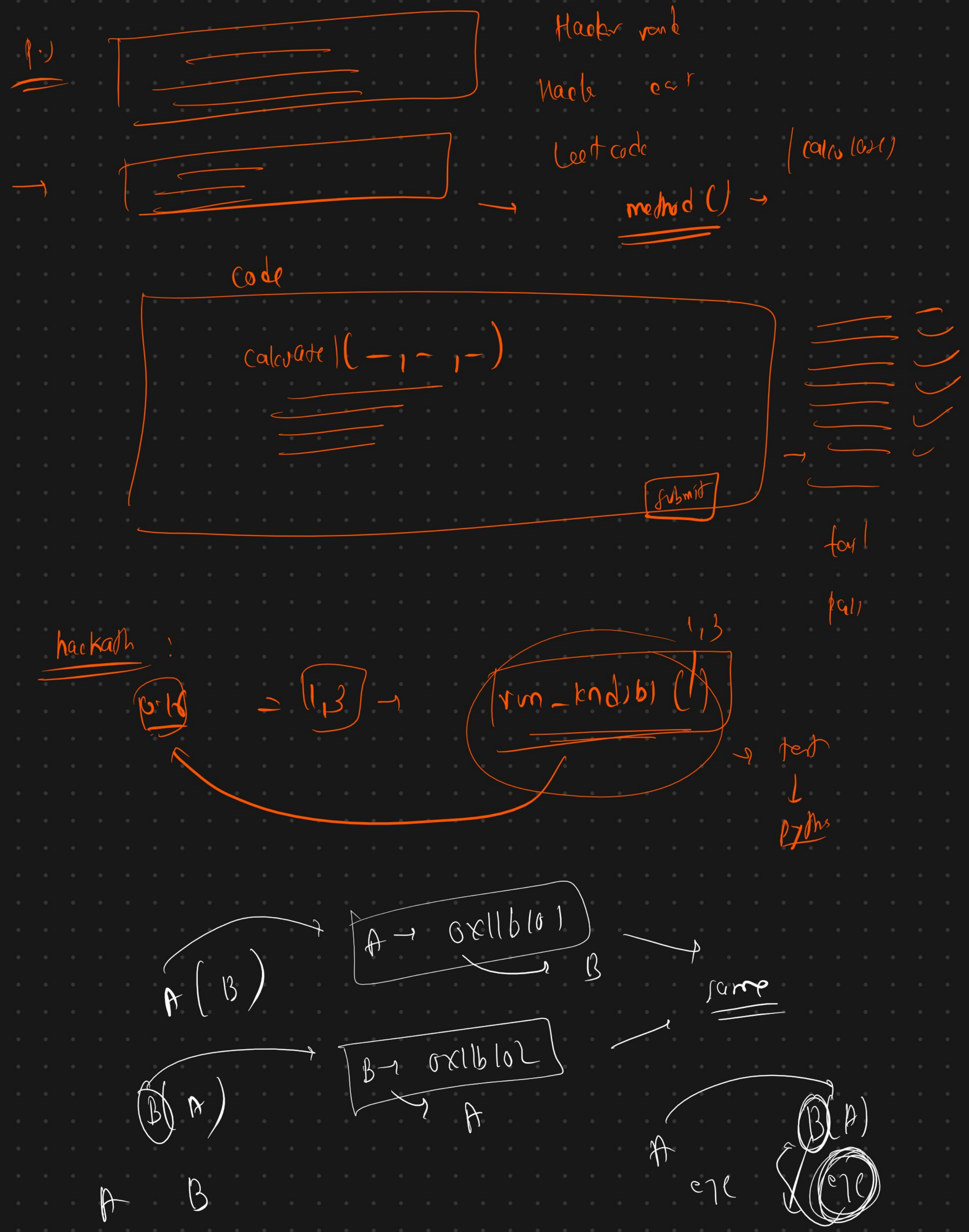


OOP →

- Inheritance (1)
- polymorphism → overriding
- Encapsulation (2)
- Abstraction (3)

OOPs → super

→ modular, code reusability
→ manage code
→ Security
→ Standard
→ Standard behaviour



Hacker rank
Leet code
method ()
code
submit
fail
pass
test
A → 0x11b101
B
A(B)
B → 0x11b102
A
same
B(A)
A B