3-05-2025

__/__/____

Lecture - 2

OOPs Pillars

History of programming :-

→ Machine language (0/1)

01000101 1100101

- Prone to error
- Tedious
- Scalable X

→ Assembley Level language

MOV A 61H

- Prone to error
- Scalable ↓ X
- Tedious

→ Procedural Programming

∴ Note :- 'C' language is one of the best example of procedural programming.

- functions
- loops
- Block (if,else / switch)

→ Do this

→ Then do this.

__/__/____

Object oriented programming :-

- Real world Modeling
- Data Security
- highly Scalable / reuseable.

Object ⟹ Interact

- Characteristics
- Behaviour

Example : Car

Character
- engine
- Brand
- Model
- Wheels

Behaviour (function)
- Start()
- Stop()
- gearShift()
- accelerate
- break

Car → Brand
→ Model
→ is Engine ON

→ Start
→ Stop
→ gearShift()

```
String brand;
String Model;
bool is Engine ON;

Start(){
  ..
Stop(){
}
```

```
String name;
void drive
(brand
Model){

Start();
gearShift()
accelerate.(
```

__/__/____


```
class Car {
    String brand;
    String model;
    bool isEngineOn;
    void start() {}
    void stop() {}
}
```

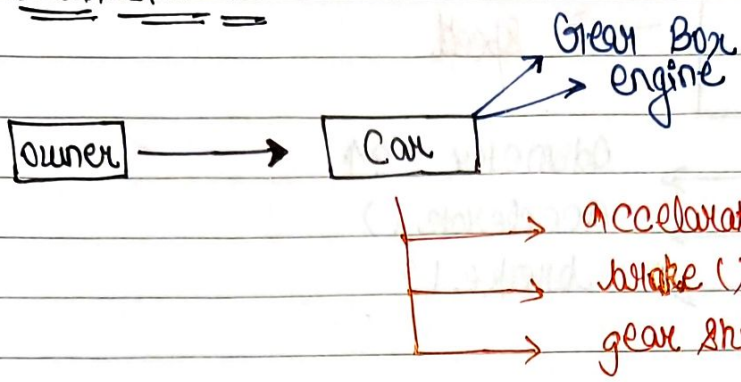
```
class Owner {
    Car car;
    String name;
    void drive() {
        car.start();
        car.shift();
    }
}
```

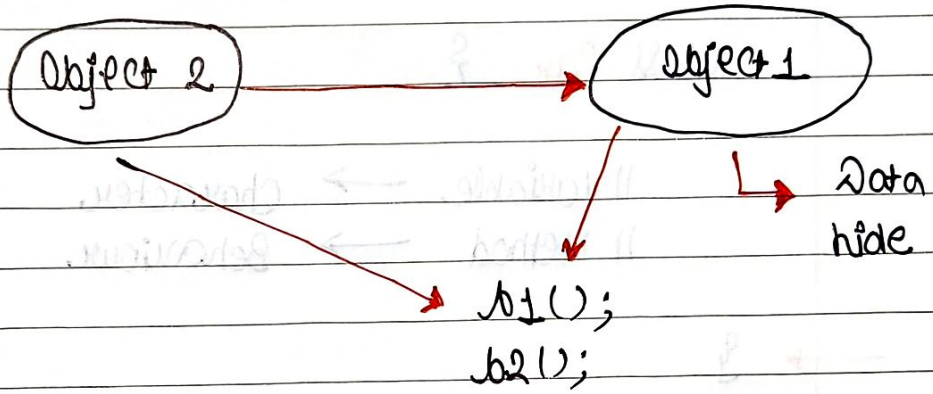
Pillars of OOPs.

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

__/__/____

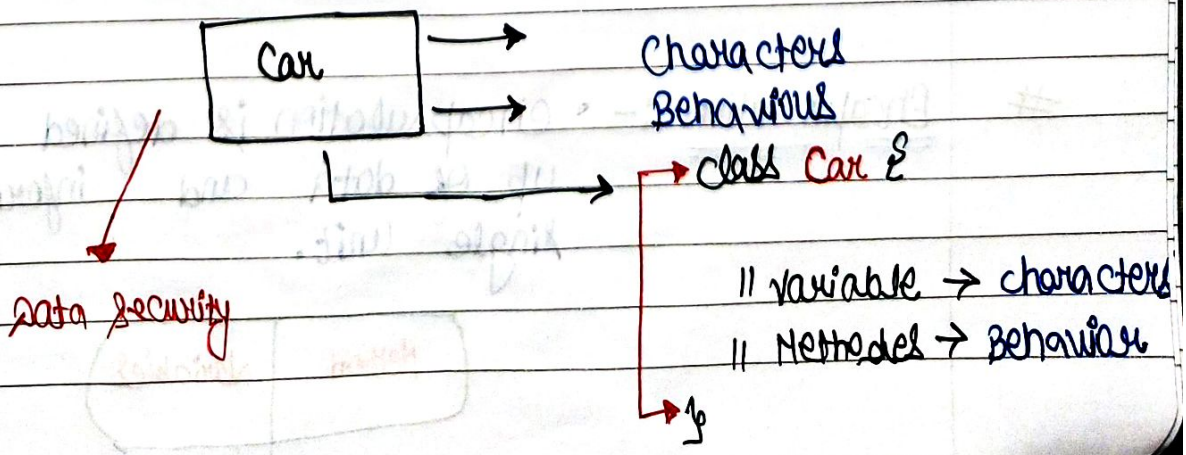
Abstraction :

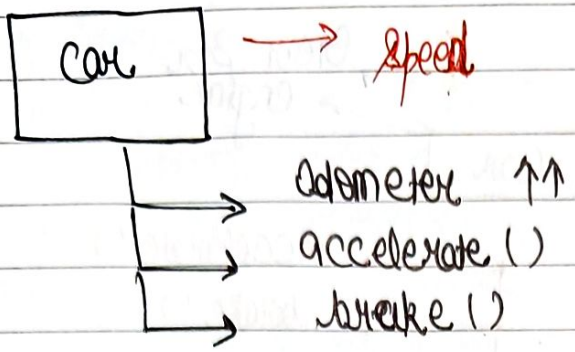




- **Abstraction :** Abstraction hides unnecessary details from a client, and showcase only what is necessary.

Encapsulation :- → Like a Capsule.



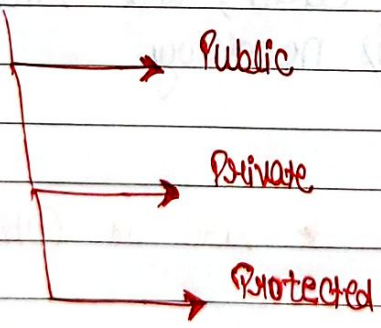


Class Car {

// variable → Character
// Method → Behaviour

→ }

Access Modifier :-



Encapsulation :- encapsulation is defined as wrapping up of data and information in a single unit.