Day :- 02 :- OOPs Real-World Application

History Of Programming:-

- Machine Language (Old)
- Assembly level Languages.
- Procedural Programming :-
  - Functions
  - Loops
  - Blocks.
- OO Programming ?

↓

• Why OO programming?

→ • Real-World Modelling
• Data Security
• Scalable / Reusable
→ These are not in the Procedural Programming.

• Objects ⇒ Interact
→ characteristics
→ Behaviour

Example: Car:- Character:- i) Engine
ii) Brand
iii) Model
iv) Wheels.

Behaviour:- i) start ()
ii) stop ()
iii) Gear shifting ()
iv) Brake ()
v) accelerate ()
vi) breare ()

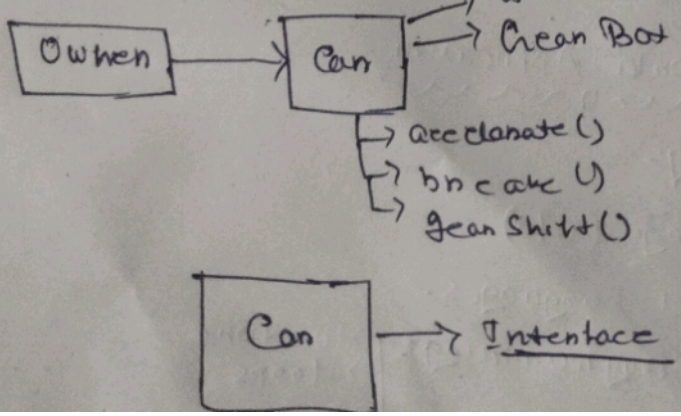
```
Class Car {
"Blueprint of the car
};
```
→ We can make instance of it.

• Idealogy of OOPs

In Object-Oriented-Programming the Object interacts with the Other Objects.

• Pillars of OOPs
→ Abstraction
→ Encapsulation
→ Inheritance
→ Polymorphism.

■ Abstraction:-



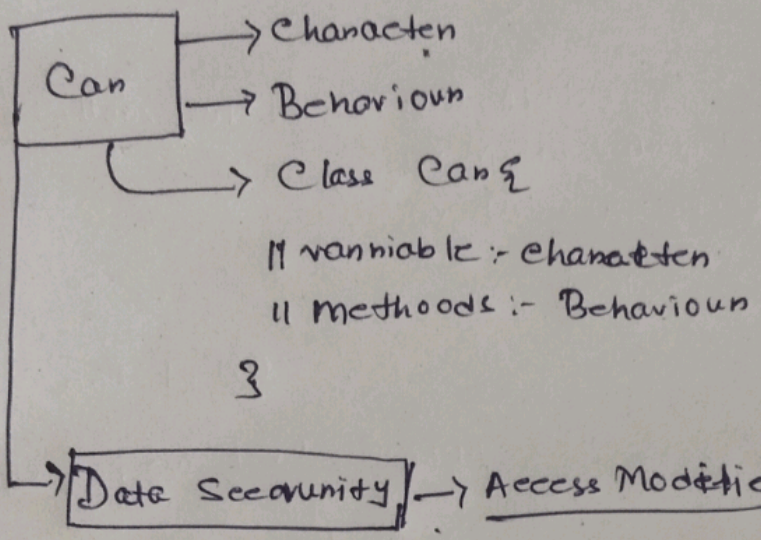
⇒ In the Real Life the Objects are Abstracted means some data are hidden there.

■ Abstraction hides unnecessary details from a client, and show-case only what is necessary.

Example:- High Level Programming

i) C++ / Java / Kotlin

■ Encapsulation :-> Live A Capsule.



=> Abstraction says something like about the engine of the car.... You don't have to know. If you also know there are no matters. But Encapsulation says there are something in the Object if you know there is a problem. Like:- Car Odometer.