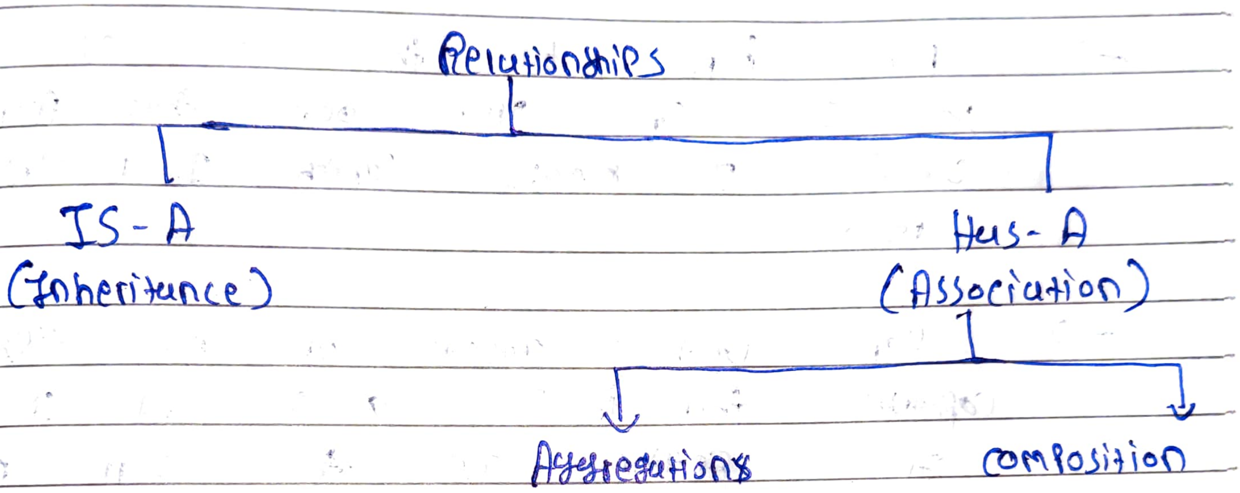
## * Relationship Between Classes.
(Important for Real-life Projects)

-> There are two types of Relationship Between classes.



## * Advantages of Relationship.

1. Code - Re-usability.
2. Cost - Cutting.
3) Reduce - redundancy (Remove un-necessory code)

-> Inheritance we covered In previous Notes
Now Association (Has - A Relation ship)

## * Association

```
class Engine {

}
class Car ~~Extends Engine~~ {
    Engine e = new Engine ();
}
```

Car $\boxed{\text{HAS - A}}$ Engine

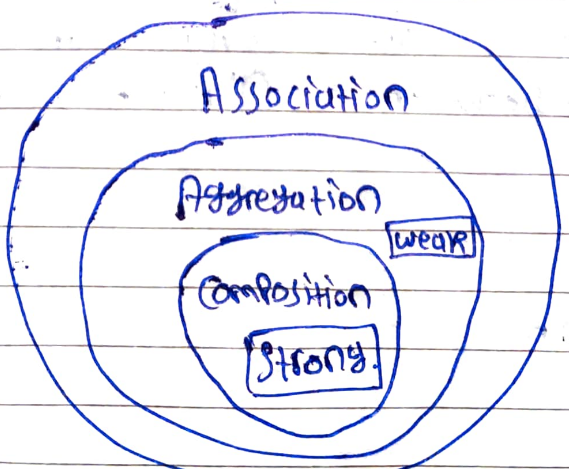
→ Here if I Inherit Engine In Car Class then All Properties of Engine Class are Extended In car Class.

→ But here by the use of e ~~sse~~ I Can use only that Properties what I need. Which I require In Project.

→ Here Both classes are not tightly copouled. then if I ~~a~~ do any changes In Engine class that will not affect on car Class.

→ Here Relation Between Both class is not blood Relation.

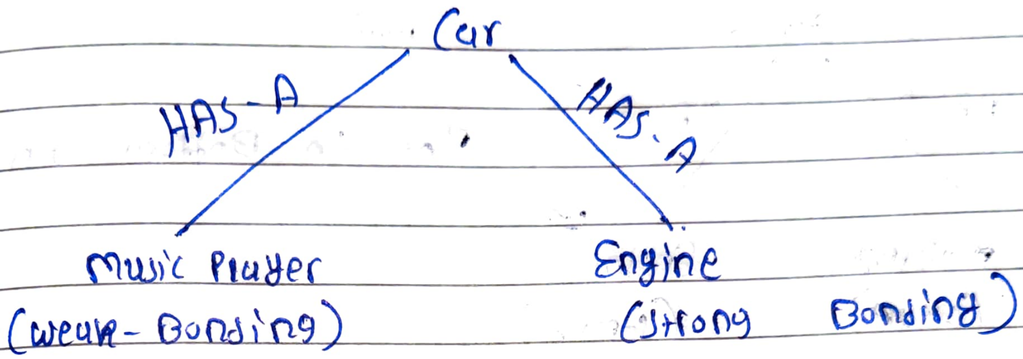
→ Association has two forms.



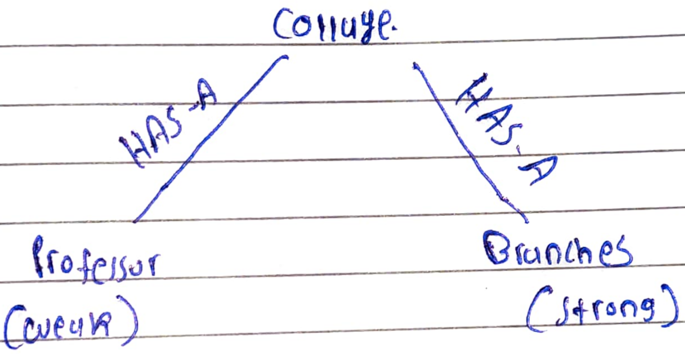
→ It a Bonding Between two classes is ~~Agg~~ ~~weak~~ weak. then it is ~~weak~~ ~~Bonding~~. Aggregation Bonding.

-) It A Bonding Between two classes is strong then it is a Composition Bonding.

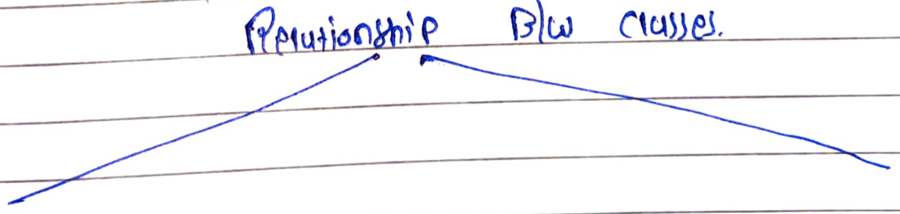
Ex:-



Ex-2 :



* Summary.



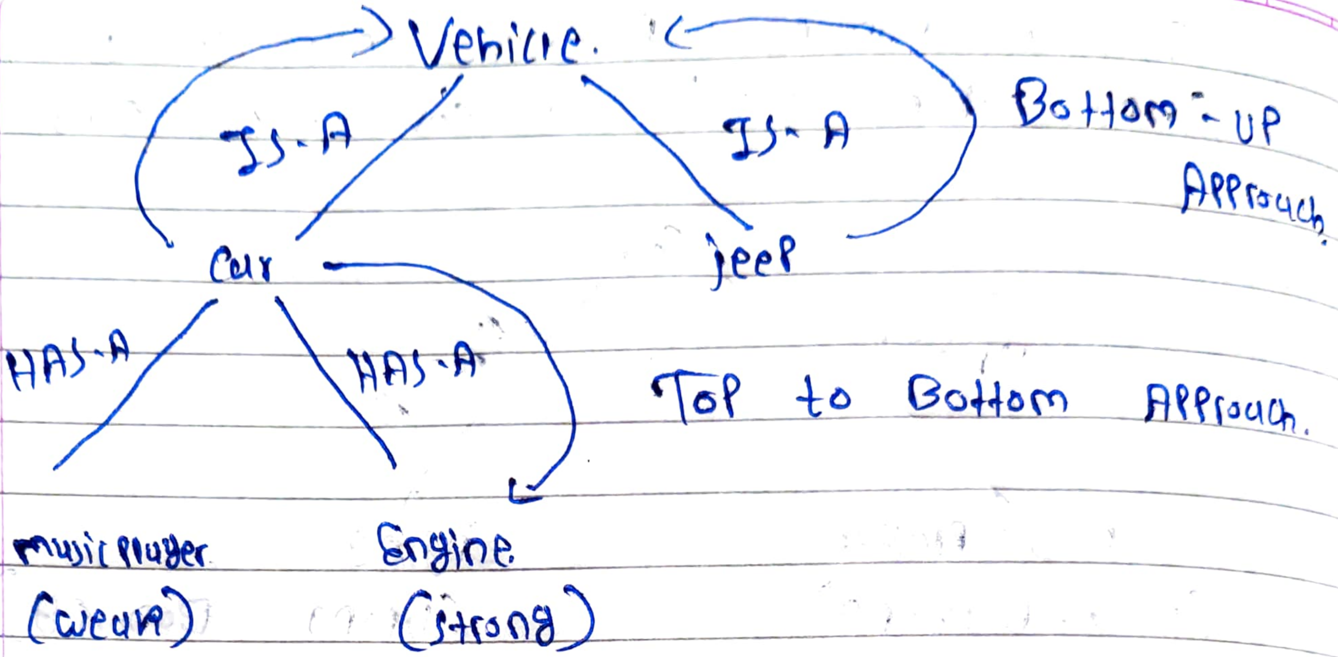
Inheritance

- (IS-A)
- Extends keyword.
- blood Relation
- tightly ~~cofouted~~. Coupled.

Association

- (HAS-A)
- ref, variable, new
- non-blood Relation.
- not tightly coupled.

-) we should use Association In large Projects.


Vehicle.
IS-A
IS-A
Bottom-UP Approach.
Car
jeep
HAS-A
HAS-A
Top to Bottom Approach.
music player
(weak)
Engine.
(strong)