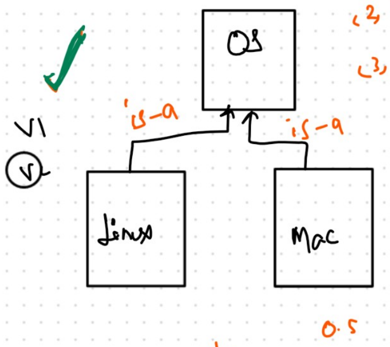
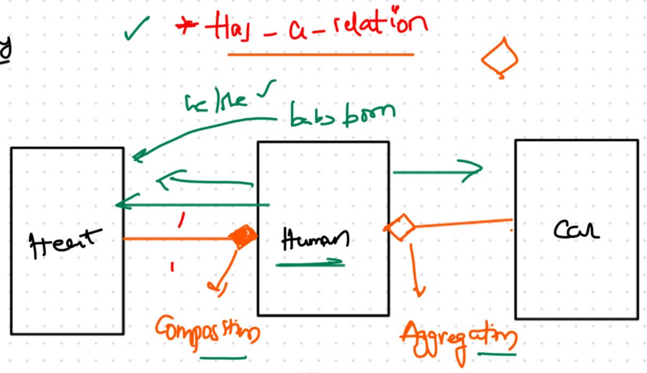
* Is_a- relation

(1) Single
(2) Multi-level
(3) Hierarchical
} Legacy



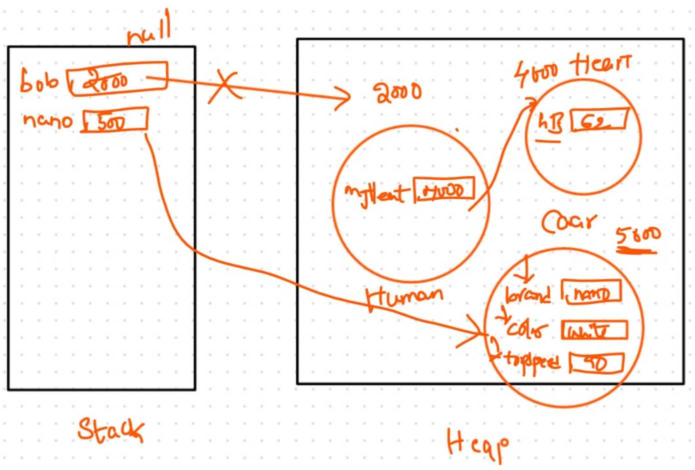
✓ * Has_a_relation



There are two types
(1) Composition
(2) Aggregation

```java
public static void main(String[] args) {
    Human bob = new Human();
    System.out.println(bob.myHeart.hearBeat);
    Caar nano= new Caar();
    bob.displayCarDetails(nano);
}
```

Memory mapping



Notes

→ In a real-world, we not only find is-a relation among objects, but also has-a-relation

→ parent_child relation is considered as "is-a relation" (extends)

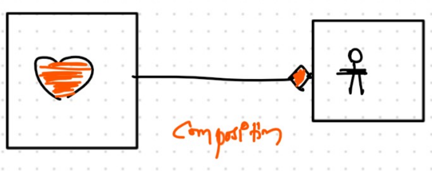
→ "has-a" relation is achieved in 2 ways
  (1) composition
  (2) aggregation

→ Composition: when the main object is destroyed, if sub objects also gets destroyed, then the objects are called a "composition"

→ Subobjects are created when the main objects are created

→ We can achieve this by making the subjects as the instance variables in the main object & creating them at constructor level

examples
(1) Human has heart
(2) Car has engine
(3) Computer has Os



Aggregation: If there is has-a-relation b/w two objects & if one object is destroyed, if the other object remains to exist, then such objects are considered as aggregate objects

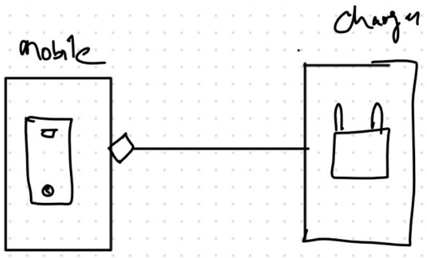
→ In this case, the objects are created seperately & they are not dependent for creation, deletion of another object.

→ We can achieve this by passing the object as parameter for a method of another class

Examples

① Human - Car

② Mobile - Charger


mobile
charger