

RELATIONSHIP



RELATIONSHIP :

The connection (Association) between two objects is known as the relationship

TYPES OF RELATIONSHIPS:

I. has-a relationship

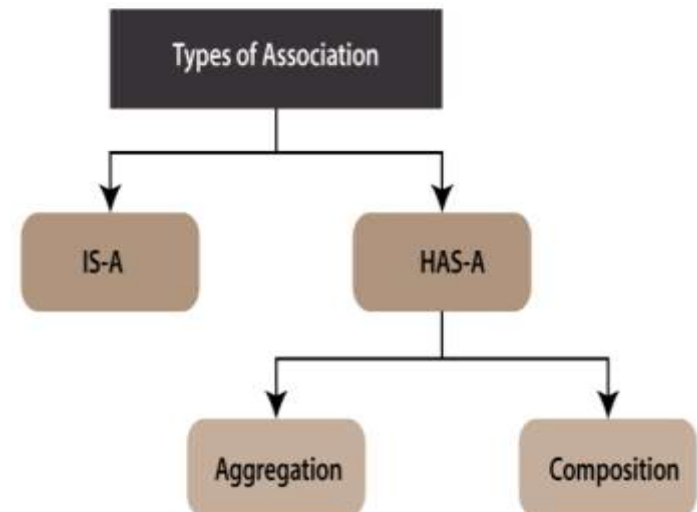
II. Is-A relationship

has-a RELATIONSHIP :

- If one object is dependent on another object it is known as a has-a relationship.
- Based on the level of dependency, a relationship is classified into two types.

I. Aggregation

II. Composition



AGGREGATION :

- The dependency between two objects such that one object can exist without the other is known as aggregation.

EXAMPLE :

Cab-Ola, Train-Online ticket booking, Bus-Passenger, etc,...

COMPOSITION :

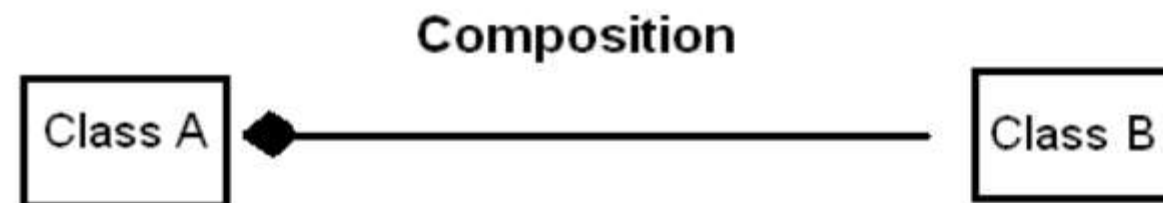
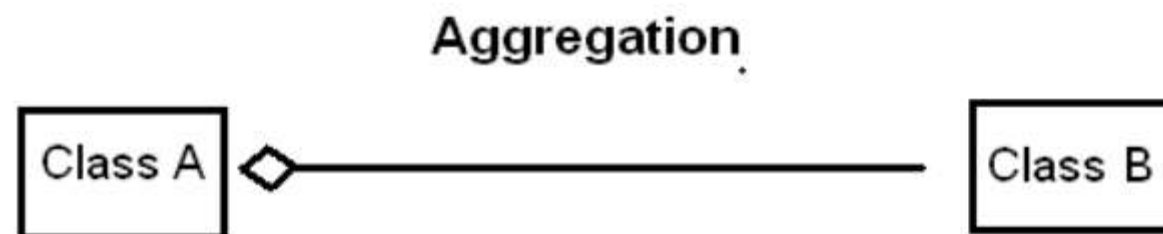
- The dependency between two objects such that one object can't exist without the other is known as composition.

EXAMPLE :

Car-Engine, Human-Oxygen, etc,...

NOTE:

We can achieve a has-a relationship in java by placing the reference variable of the dependent class inside another class.



The instance of a dependent object can be created in two ways

1. Early instantiation

2. Late / Lazy instantiation

EARLY INSTANTIATION :

- If the instance of a dependent object is created implicitly it is known as early instantiation.
- This design can be achieved with the help of an initializer.

STEPS TO ACHIEVE EARLY INSTANTIATION :

STEP 1: Create a dependent class.

STEP 2: Create another class and place the reference of the dependent object variable inside the class.

STEP 3: Create a constructor for the class which also accepts the dependent type object.

STEP 4: Create the object for a class so the object of a dependent object is also created.

EXAMPLE :

Whenever we buy a car, the engine is by default mounted inside the car. So the engine object should be created inside the car object.

class Engine

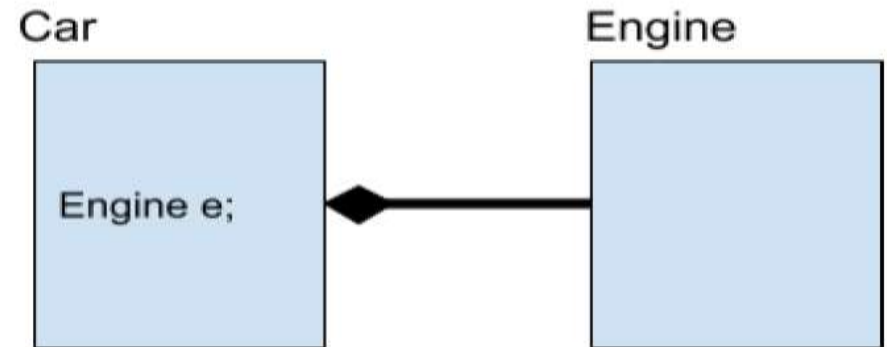
```
{  
    int eno = 123;  
}
```

class Car

```
{  
    Engine e = new Engine();  
}
```

Car c = new Car();

S.o.pln(c.e.eno); //getting engine no by using car object reference



LAZY / LATE INSTANTIATION :

- In this design, the instance of the dependent object is created only when it is required (It is not implicitly created).
- We can achieve this design with the help of a method, it can be called a helper.

STEPS TO ACHIEVE LATE / LAZY INSTANTIATION :

STEP 1: Create a dependent class.

STEP 2: Create another class and define a parameterized method that will accept the reference of the dependent object and inside that method initialize the dependent object.

STEP 3: Create the object for a class and call a method by passing dependent type object reference so that we can achieve a late/lazy instantiation.