Inheritance:

It is inherting the properties of parent class into child class OR,

Inheritance is the procedure by which one object acquires all the properties and behaviours of parent object.

Note: We achieve inheritance through extends keyword.

```
class Animal { parent class, super class }
{
    void eat()
    {
        S.O.P("I am eating");
    }  child class, sub class
}
class Dog extends Animal
{
    psvm()
    {
        Dog d = new Dog();
        d.eat();
    }
}
```

Dog IS-A Animal

Dog → child class. Animal → parent class

=) Sparrow IS-A Bird.

=) Car IS-A Vehicle.

=) Surgeon IS-A Doctor.

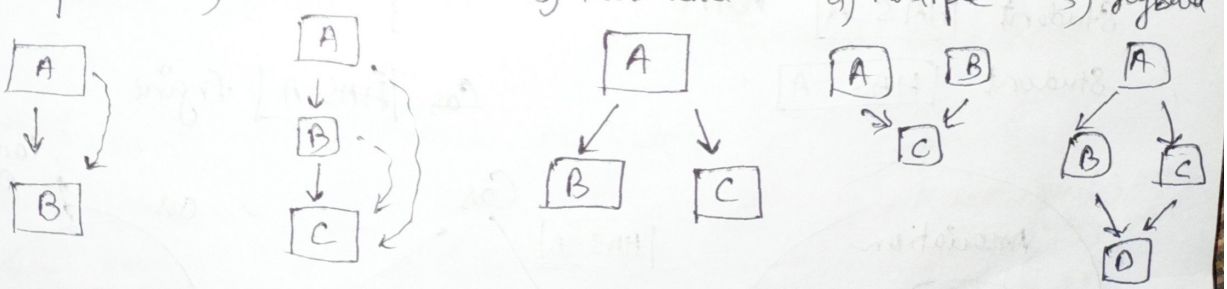
---

(1) Code reusability

(2) We can achieve polymorphism using inheritance.

Types of Inheritance:

1) Single 2) Multilevel 3) Hierarchical 4) ~~Multiple~~ 5) ~~Hybrid~~ (not available to Java)



When cannot a class be inherited?

1) If child class tries to inherit constructor of parent class then this will throw an error.

2) If any method is private in parent class then also it cannot be inherited.

class A
{
}
class B
{
}

Types of Relationship b/w classes

- (IS-A) Inheritance
- Association (HAS A)
  - Aggregation
  - composition

Advantages :

1) code reuseability.
2) cost cutting.
3) reduce redendancy.

Inheritance (IS-A)

```
class Vehicle
{
}
class Car extends Vehicle
{
}
```

Car IS-A Vechicle.

Association (HAS-A) :

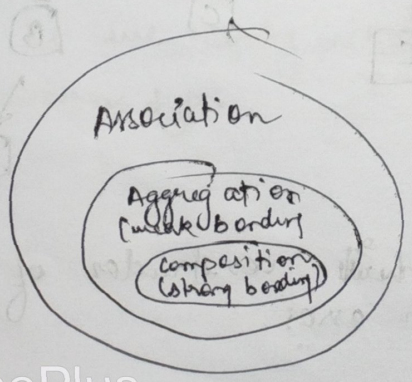
```
class Student
{
String name;
int rollno;
}
```

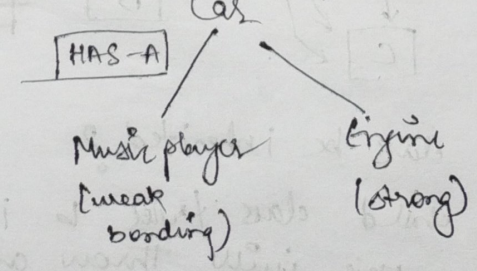
Student HAS-A name

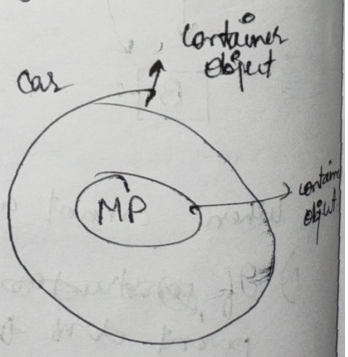
Student HAS-A .

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

Car HAS-A Engine







College

HAS-A — professor (weak)

HAS-A — Branches (CSE, IT, ME) (Strong)

Relationship b/w classes:

- Inheritance (IS-A)
  - extends
  - blood relation
  - tightly coupled
- Association (HAS-A)
  - ref variable, new
  - non blood relation
  - not tightly coupled
  - Aggregation (weak)
  - Composition (Strong)