

```
class Date{
}
```

```
class Vehicle{
}
```

```
class Employee{
    Date doj; // Association
    Vehicle veh; // Association
```

```
Employee(){
    this.doj = new Date();
}
```

```
Employee(int,String,double,int,int,int){
    this.doj = new Date(int,int,int);
}
```

```
Employee(int,String,double,int,int,int,Vehicle){
    this.doj = new Date(int,int,int);
    this.veh = vehcile;
}
```

```
accept(){
}
}
```

```
superclass{
    method1();
}
```

```
Subclass extends Superclass{
    method1();-> override
}
```

```
Subclass s1 = new Subclass();
s1.method1();
```

```
Superclass s2 = new Subclass(); // upcasting
s2.method1();
```

```
if(s2 instanceof Subclass)
    Subclass s3 = (Subclass) s2; /// downcasting
```

```
String toString(){
    return "Date = "+this.day + "/" +this.month
}
```

Inheritance

is-a realtionship

Parent class -> Base class -> Superclass

Child class -> Derived class -> sub class

Types of inheritance

single
multilevel
hierarchy
hybrid

multiple implementation - NOT supported
multiple interface - Supported

FQCN@hashcode



