Inheritance

ICC Board
FIFA Board

① ②

Client - I

class Cricketer{   s/w - I
- name
- age
- height
- weight
- matches played
- strike rate
- 100's
- 50's
- wickets
- category
- manOfTheMatch

}

3cr
6 mnths

class Footballer{   s/w - II
- name
- age
- height
- weight
- matches played
- goals
- redcard
- penalties
- position
- offside

}

3cr
6 mnths

12 mnths
∴ 1 year

Client - 2

s/w - I

class Human {
- name
- age
- height
- weight

}

3 mnths

s/w - II ①
class Cricketer{
- matches played
- strike rate
- 100's
- 50's
- wickets
- category
- manOfTheMatch

}

3 mnths

6 mnths
8cr

s/w - III ①
class Foot {
- matches playe
- goals
- redcard
- penalties
- position
- offside

}

3 mnths
6cr

Season

	s/w1	s/w2	Earn	12m
client-1	6m	6m	6cr	
Client-2	6m	3m	9cr	3m

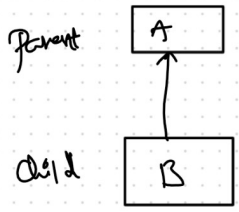
Quality
∴ quality

def: It is a mechanism where one object acquires features [properties/fields] & behaviour i.e methods of the parent object

Types of Inheritance:
1. Single Inheritance
2. Multi-level
3. Hierarchical

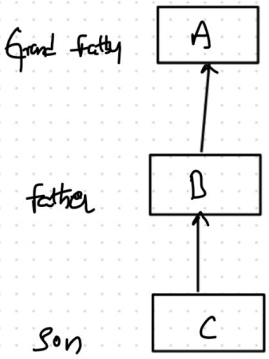
Possibilities

Single



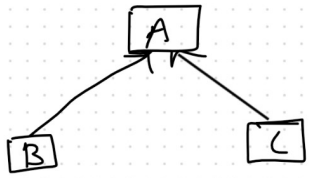

```
class A{
   ==
}

class B extends A {
   ==
}
```

Multi-level




```
class A {
   ==
}

class B extends A{
   ==
}

class C extends B{
   ==
}
```

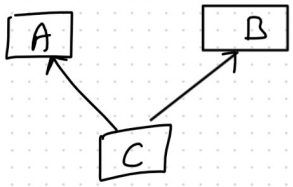
Hierarchical




```
class A{
   ==
}

class B extends A{
   ==
}

class C extends A{
   ==
}
```

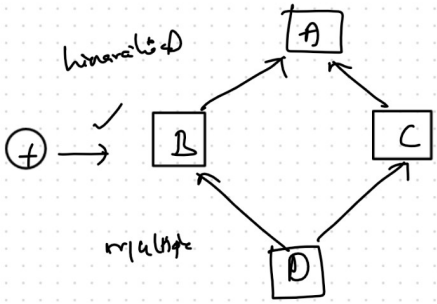
Not Possible

Multiple



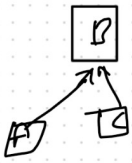

```
class A{
   ==
}

class B{
   ==
}

class C extends A, B {
   ==
}
```

Hybrid



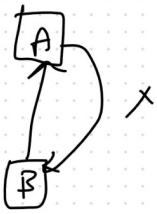

```
class A{
   ==
}

class B extends A { }

class C extends A { }

x class D extend B, C { }
```

Cyclic



$A \to B = B \to A$

$A \to B = B \to C$




```
class B extends A {
   ==
}

class A extends B {
   ==
}
```

Rules of Inheritance

(A) extends ↑

1. We have to use "extends" keyword to inherit the properties & behaviours from super class to sub class

2. "private members" do not participate in heritance

<u>example:</u>

super class

```
class A {
    int x;
    private int y;
    private void sleep() {
        ≡
    }
    public void eat() {
        ≡
    }
}
```

```
class B extends A {
    x
    eat()      } inherited except private members i.e y & sleep()
}
```

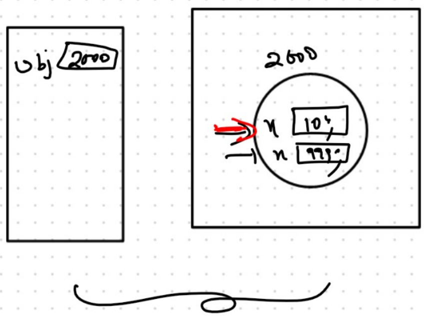
3. In case of constructors, they are not allowed to participate in heritance

4. Multiple Inheritance, Hybrid, cyclic Inheritance is not allowed for classes in Java.

5. We can have variables (instance variables) of the same name in super class as well as sub class, if we create an object for that subclass then memory will be allocated for the both the variables within the object

```
class A {
    int x = 10;
}
```

```
class B extends A {      (B → subclass, A → superclass)
    int x = 999;
    void display() {
        S.o.p(this.x);
        S.o.p(super.x);
    }
}
```



Both the variables are present

```
class Main {
    p.s.v.m(String args[]){
  →     B obj = new B()
  →     obj.display()
    }
```

⑥ to access the variable present in the super class we have to make use of "super" keyword & to access variable present in sub class we can make use of "this" keyword

⑦ Constructor doesn't participate in heritance, but the parent class constructor can be called from the child class with the help of "super()" this()

```
class Parent {

    Parent() {
        s.o.p ("parent Constructor");
    }

}
class Child extends Parent {

    child() {            Parent() {
    super()                  X
    s.op ("child constructor");
    }

class Main {

    p.s.v.m (S [] args) {
        new child();
    }
}
```

o/p

parent Constructor
Child Constructor

Case L

```
class A {
    A() {
        s.o.p ("parent constructor")
    }
}

automatically

class B extends A {
    p.s.v.m (strings args[]) {
super()     new B ()
} }
```

Note: If there is no constructor in child it will automatically calls the default constructor & default super() call is called to parent because it extends parent class

---