1️⃣ Inheritance

1. INHERITANCE

Inheritance is a mechanism in Java where one class (called the **child** or **subclass**) inherits the properties and behaviors (fields and methods) of another class (called the **parent** or **superclass**).

Main.java

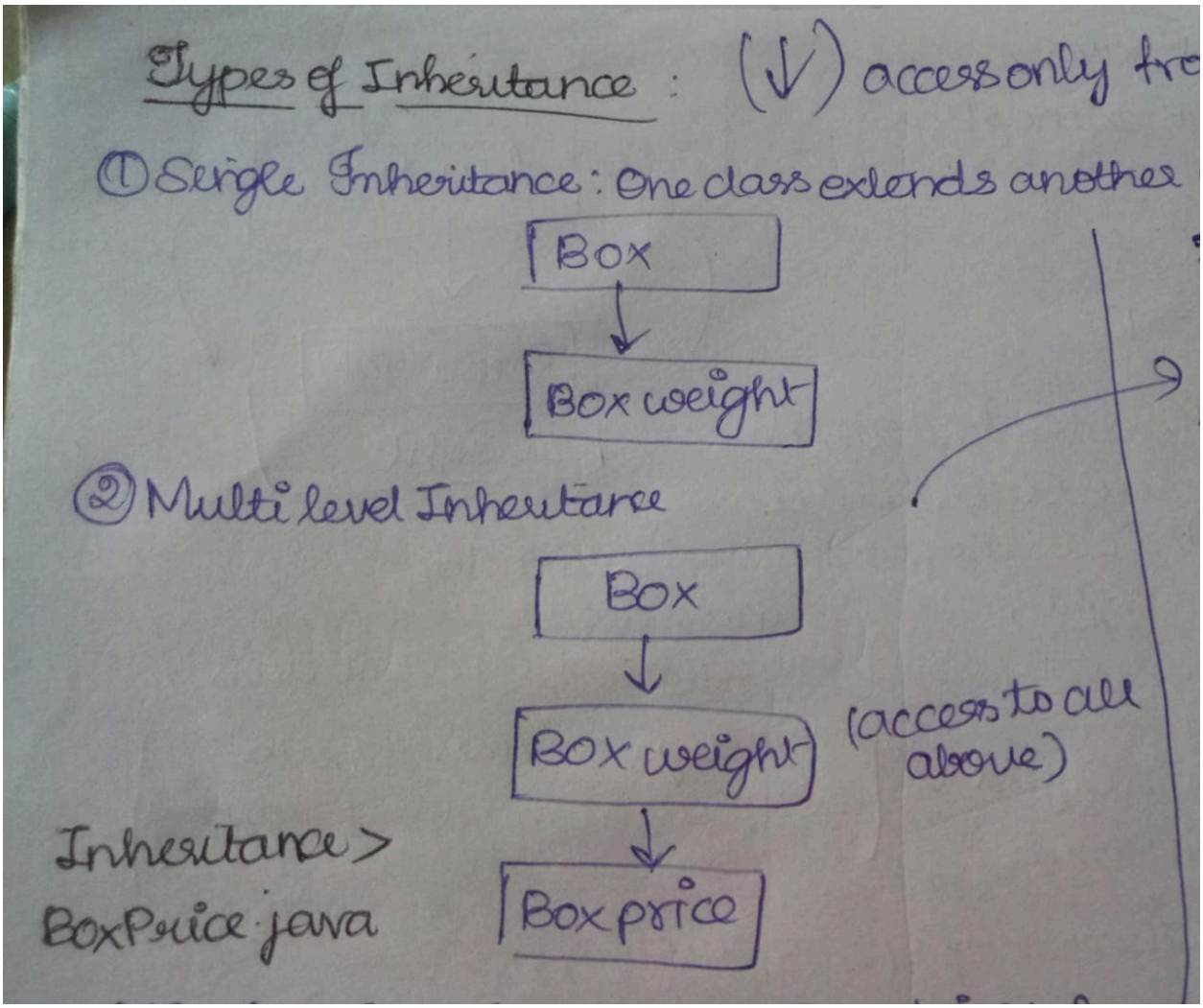
```
package Inheritance;
public class Main{
    public static void main(String[] args){
        // Parent from Parent Inheritance
        //Initialize with the constructors based on parameters used in Object Creatic
        Box box1=new Box();
        Box box2=new Box(4);
        Box box3=new Box(4,5,6);
        Box box4=new Box(box2);
        System.out.println("Box Width: "+box1.w+"|| Box Length: "+box1.l+"|| Box H
        System.out.println("Box Width: "+box2.w+" || Box Length: "+box2.l+" || Box
        System.out.println("Box Width: "+box3.w+" || Box Length: "+box3.l+" || Box
        System.out.println("Box Width: "+box4.w+" || Box Length: "+box4.l+" || Box

        //Child from Child Inheritance
        BoxWeight box5=new BoxWeight();
        BoxWeight box6=new BoxWeight(4,5,6,7);
        System.out.println("Box Width: "+box5.w+" || Box Length: "+box5.l+" || Box
        System.out.println("Box Width: "+box6.w+" || Box Length: "+box6.l+" || Box

        //Parent from Child Inheritance
```

```
        Box box7=new BoxWeight(2,3,4,5);
        // System.out.println("Box Weight: "+box7.weight); // X(Not Allowed ...Acces
        System.out.println("Box Length: "+box7.l); // Allowed ... Since variable is def

        //Child from Parent Reference
        // BoxWeight box8=new Box();
        // System.out.println("Box Length: "+box7.l);// Allowed....

        // Multiple Inheritance Example
        BoxPrice box9=new BoxPrice();
        BoxPrice box10=new BoxPrice(5,8,200);
        System.out.println("Box Price: "+box9.cost);
        System.out.println("Box Price: "+box10.cost);

        // Hierarchial Inheritance
        BoxColor colorBox = new BoxColor(2, 3, 4, 5, 100, "Blue");
        System.out.println("Box Color: " + colorBox.color + " | Cost: " + colorBox.co



    }
}
```

Box.java(Parent Class)

```
package Inheritance;

public class Box{
    double l;
    double h;
    double w;
    //default constructors when no parameters are passed
    Box(){
        this.l=-1;
        this.h=-1;
        this.w=-1;
    }
```

```
    // Constructor when only one parameter
    Box(double side){
        this.h=side;
        this.l=side;
        this.w=side;
    }

    //Constructor when length, breadth and height parameters are passed
    Box(double w, double l,double h){
        this.l=l;
        this.h=h;
        this.w=w;
    }

    //Constructor using another object as a parameter and Initialized using the valu
    Box(Box old){
        this.h=old.h;
        this.l=old.l;
        this.w=old.w;
    }
}
```



BoxWeight.java(Single Inheritance)

```
package Inheritance;

public class BoxWeight extends Box{
    double weight;
    public BoxWeight(){
        this.weight=-1;
    }

    public BoxWeight(double w, double l, double h,double weight){
        super(w,l,h);
        this.weight=weight;
```

```
    }
    public BoxWeight(double side, double weight){
    super(side);
    this.weight = weight;
    }
}
```

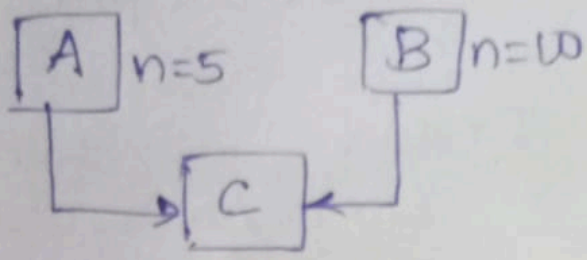
BoxPrice.java(Multi-level Inheritance)

```
package Inheritance;
public class BoxPrice extends BoxWeight {
    double cost;

    BoxPrice() {
        super();
        this.cost = -1;
    }

    BoxPrice(BoxPrice other) {
        super(other.w, other.l, other.h, other.weight);
        this.cost = other.cost;
    }

    BoxPrice(double l, double h, double w, double weight, double price) {
        super(l, h, w, weight);
        this.cost = price;
    }

    BoxPrice(double side, double weight, double cost) {
        super(side, weight);
        this.cost = cost;
    }
}
```

BoxColor.java(Hierarchial Inheritance)

③ Multiple Inheritance

One class extending more than one classes



C obj = new C();

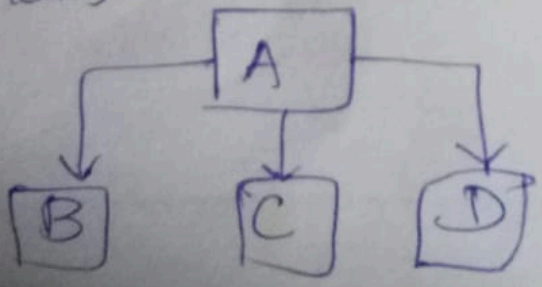
Use case C.n// ? causes an confusion

[Multiple Inheritance - Not allowed in Java]

↓ // Abstract classes & Interfaces

④ Hierarchical Inheritance

One class is inherited by many classes

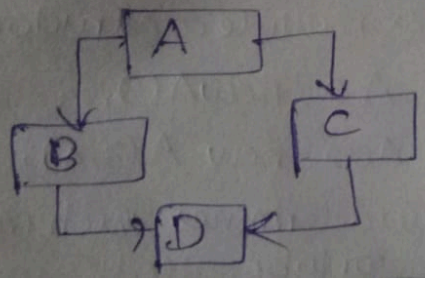


```
package Inheritance;

public class BoxColor extends BoxWeignt{
    String color;
    BoxColor(){
        super();
        this.color="white";
    }
    BoxColor(BoxColor other){
        super(other.w,other.l,other.h,other.weight,other.cost);
        this.color=other.color;
    }

    BoxColor(double w, double l, double w, double weight, double price,String colo
        super(w,l,h,weight,price);
        this.color=color;
    }

    BoxColor(double side, double weight, double cost,String color){
        super(side,weight,cost);
        this.color=color;

    }
}
```

⑤ Hybrid Inheritance:

combination of single & multiple inheritance

Not in Java (Infaces contd ..) Same Usecase



{ achieve this sort of functionality in interfaces }