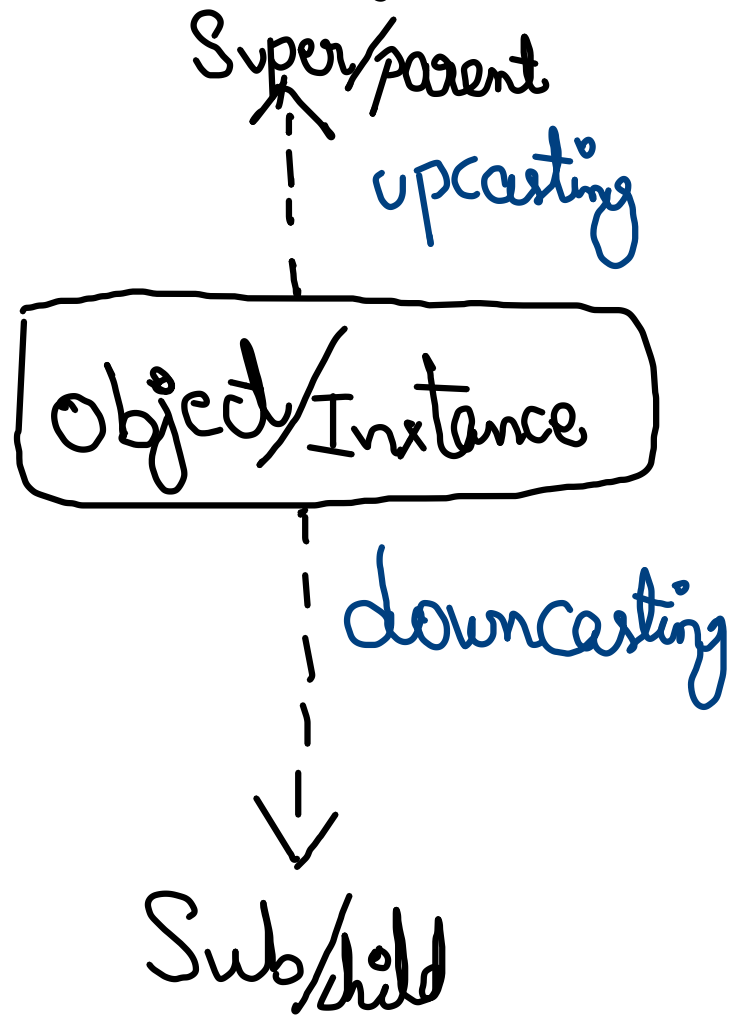


	Datatype	→ Variable = value;
<u>Prim</u>	Datatype Keywords	→ [Var Name] = Literal Value;
<u>Non Prim</u>	Class	→ Object reference variable = Instance

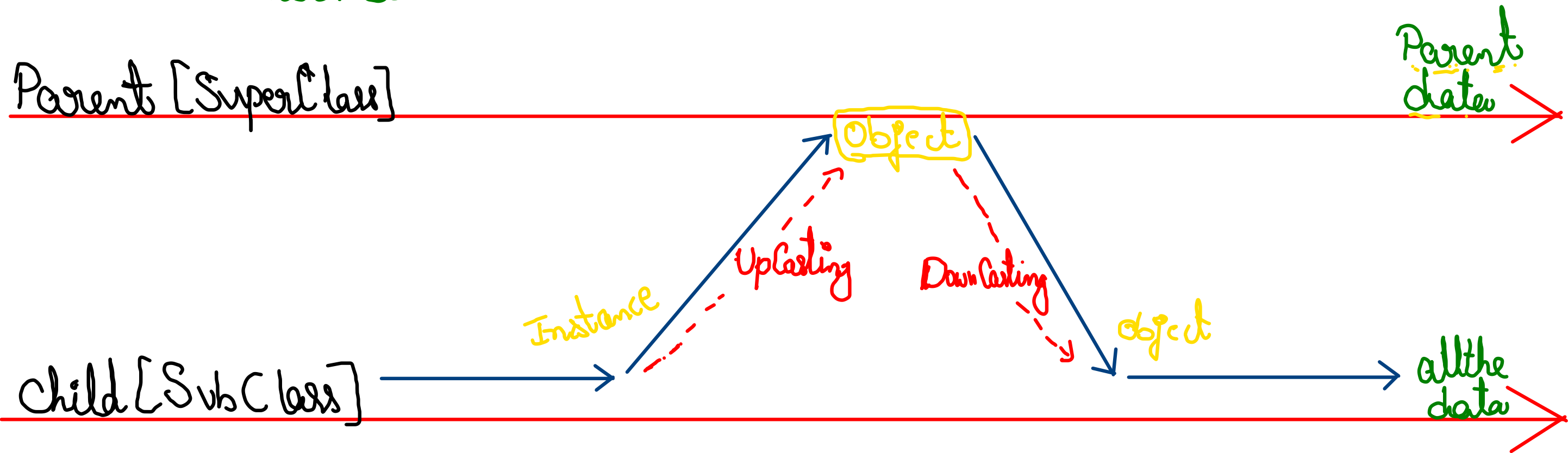
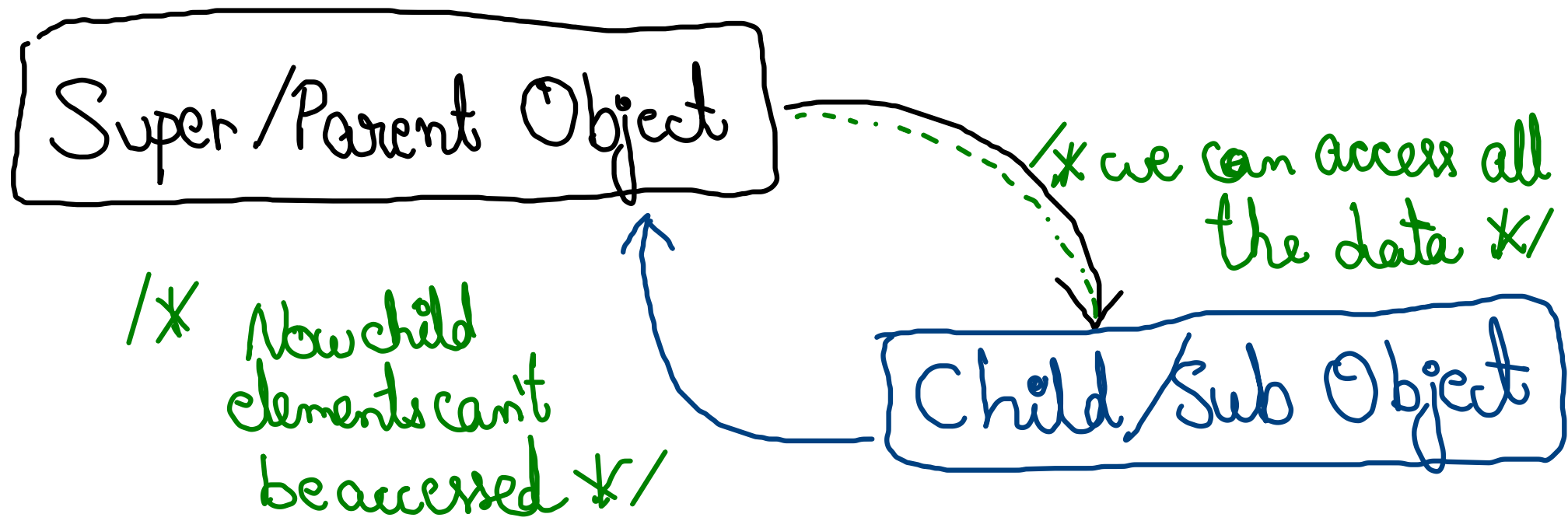
Objects Type Casting / Non Primitive

an object can either be changed to subtype or supertype.

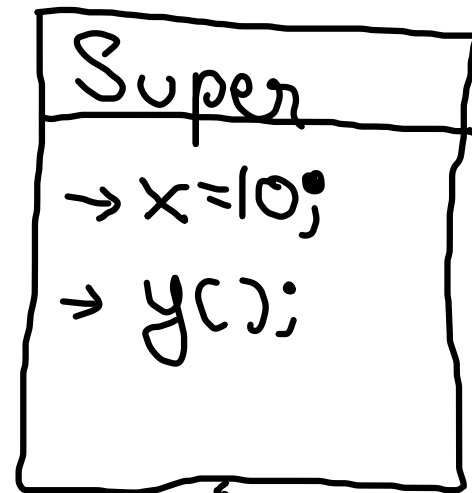


i) Up casting: Converting a child object into parent is up casting & it is implicit.

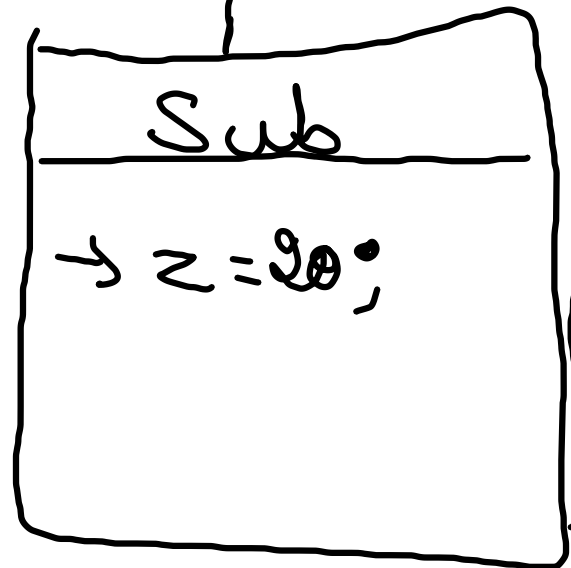
ii) Down casting: Converting a parent object into child is down casting & it is explicit.



Inheritance



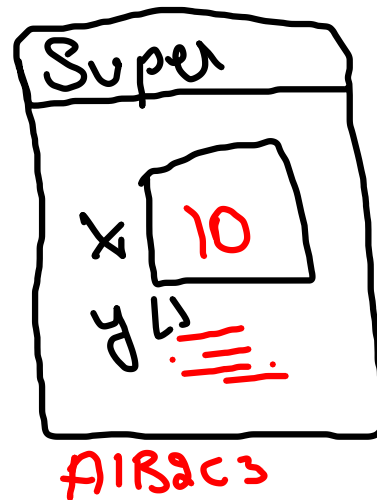
extends



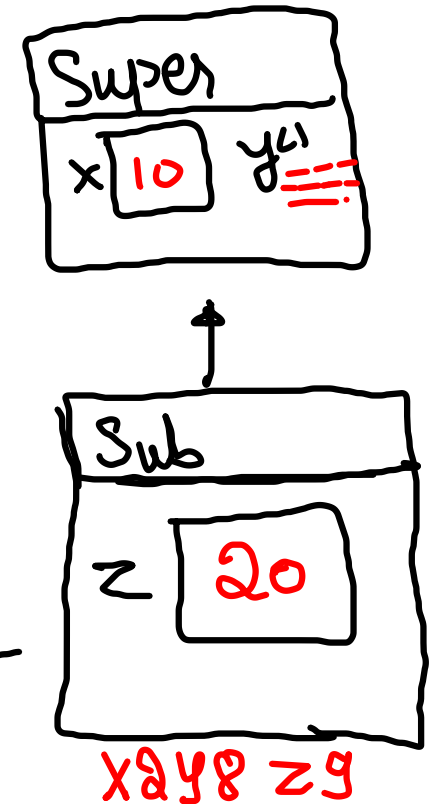
Super obj 1 = new Super(); /* parent instance only for parent data */

Sub obj 2 = new Sub(); /* child instance for both parent & child data */

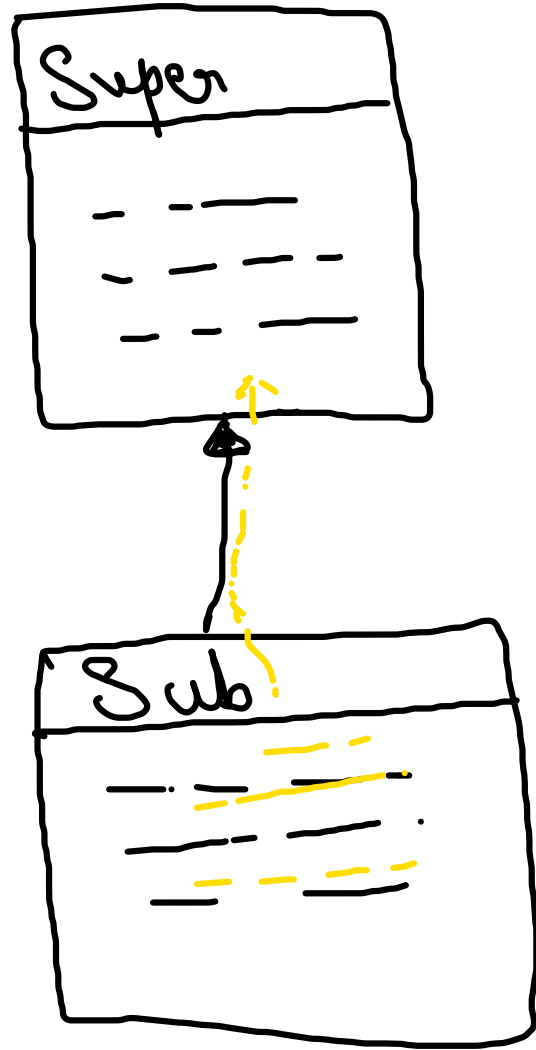
Obj 1 ⇒



Obj 2 ⇒



UpCasting



```
Sub obj1 = new Sub();
```

```
Super obj2 = obj1;
```

Parent object ← child object
upcasted

→ a child instance is now moved to parent object variable & now cannot access child data.

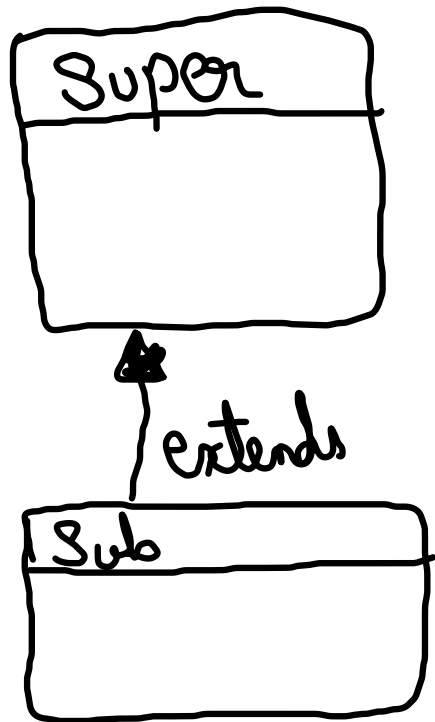
→ upcasting is implicitly performing

obj2 => Parent instance

Obj1 => { Parent instance
child instance }

DownCasting [explicitly]

→ downcasting can be performed only once parent is casted from child

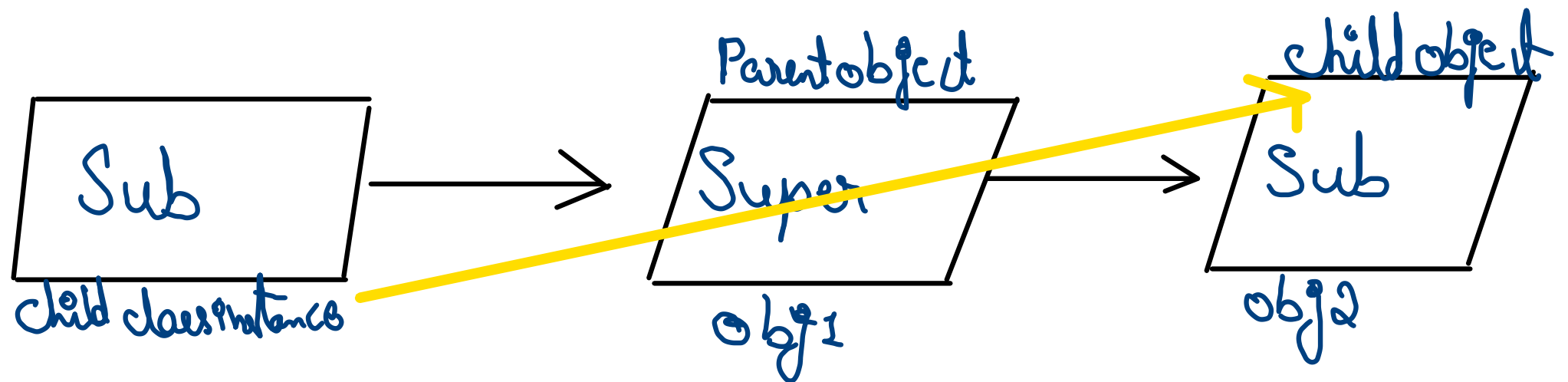


i) Upcasting: Super obj1 = new Sub();

parent (under Super)
child (above new Sub)

ii) DownCasting: Sub obj2 = (Sub) obj1;

child (under Sub)
parent (above obj1)



```

class Parent_Example{}

class Child_example extends Parent_Example {}

public class Downcasting { //driver
    public static void main(String[] args)
    {
        /*
        System.out.println("Program started");
        Child_example obj = (Child_example) (new Parent_Example());
        System.out.println( obj );//Run Time Error
        System.out.println("Program ended");
        */

        /* Runtime Error
        Parent_Example obj = new Parent_Example();
        Child_example obj2 = (Child_example) obj ;
        */

        Parent_Example obj1 = new Child_example() ; //upcasting
        Child_example obj2 = (Child_example) obj1;//downcasting

        System.out.println(obj1+"\t"+obj2);

    }
}

```

