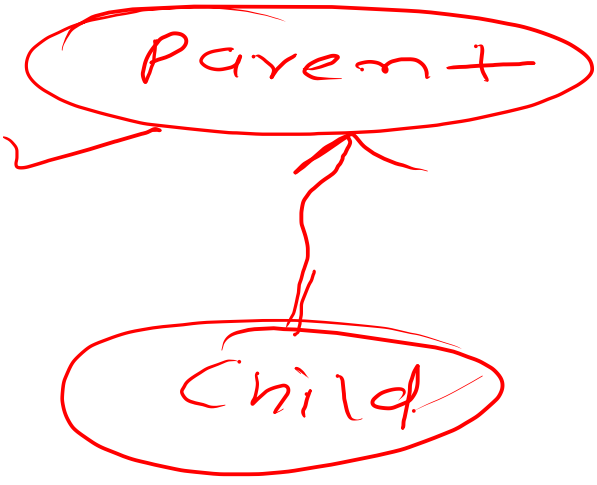
① single



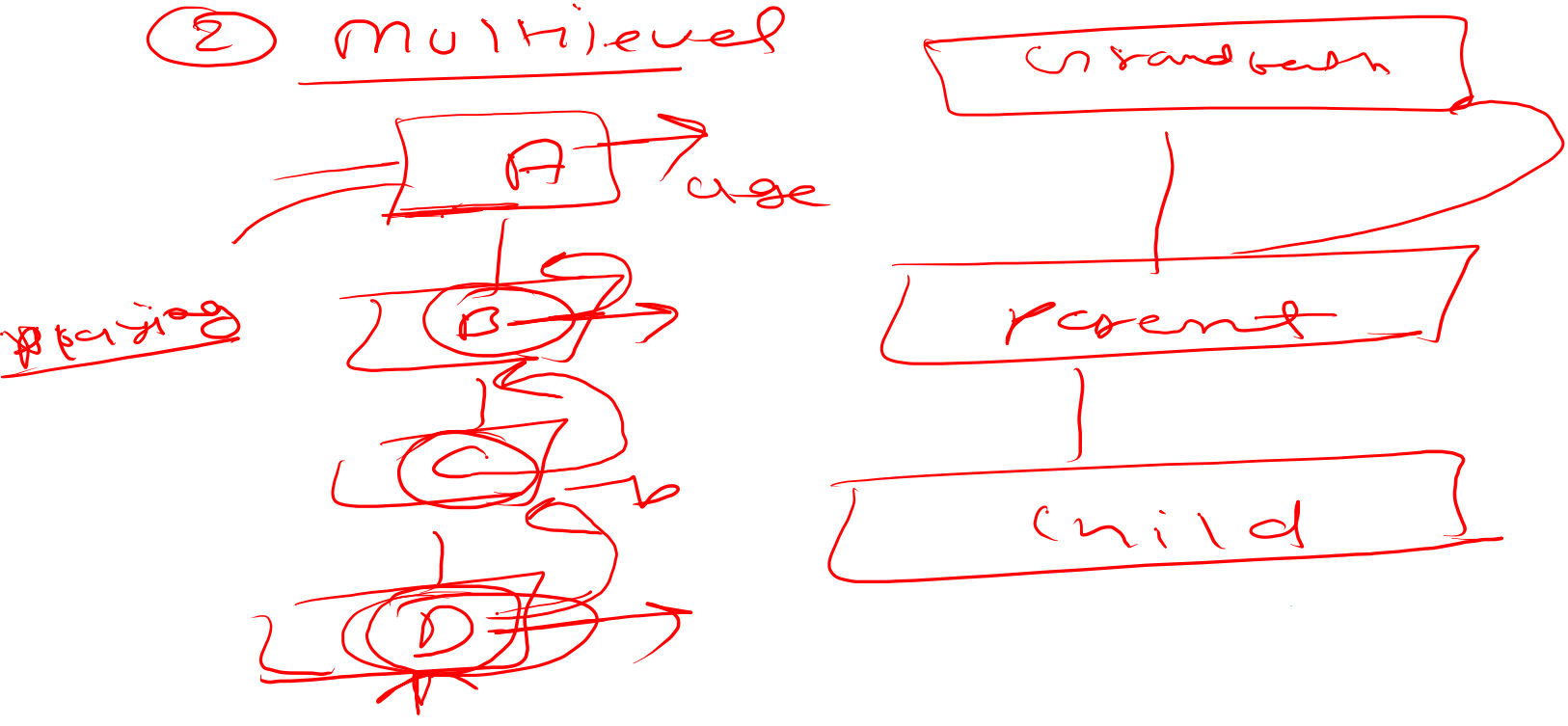
walking()
{
}

Child extends Parent {

Dancing();

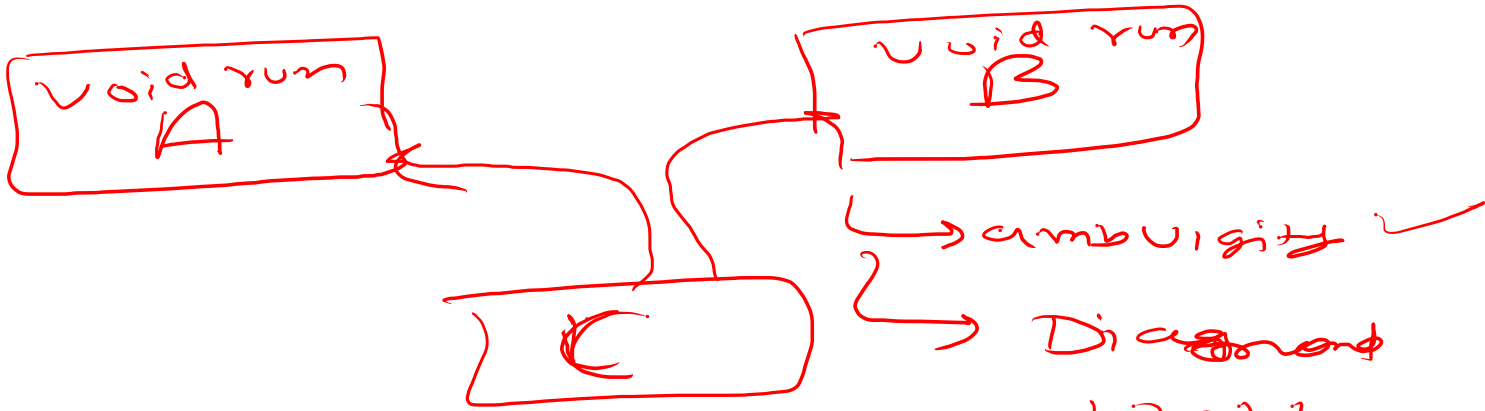
}

① Rev


2 Multilevel
A
age
B
C
D
Grandfather
Parent
Child

3 Mutiple Inheritance

It is possible by the concept of Interface.



⑤ Hybrid

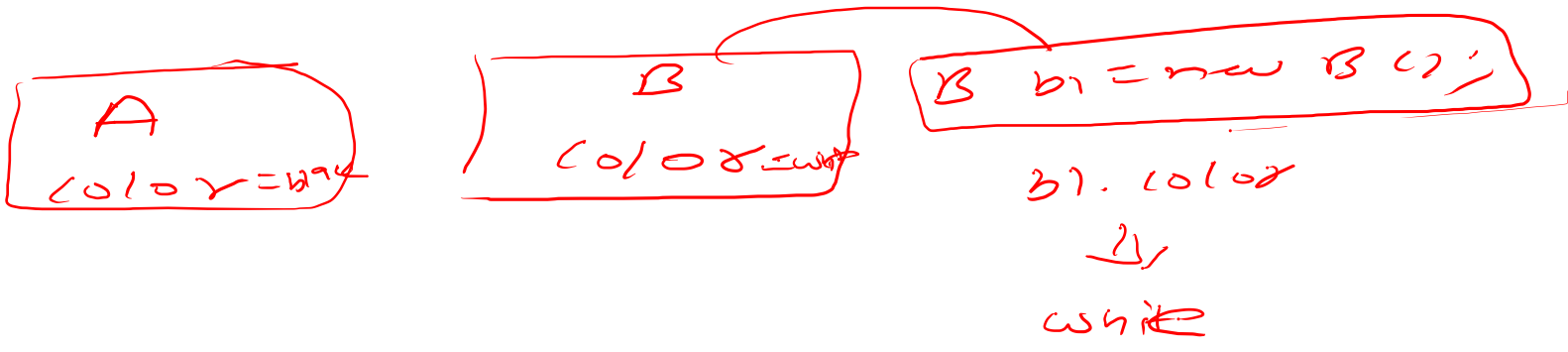
It a mixture of single level and multiple Inheritance.

Super keyword

- variable
- methods
- constructors

```
class A {
    String color = "Black";
}
class B extend A {
    String color = "white";
}
B obj = new B();
System.out.println(b.color);
```

When we want to call the parent method, properties, constructor then we are going to use the super keyword.



super By Method

```java
public class fullstack {
    1 usage  1 inheritor
    static class Frontend{
        2 usages  1 override
        void skill(){
            System.out.println("I know html css js");
        }
    }
    2 usages
    static class Backend extends Frontend{
        2 usages
        void skill(){
            super.skill();
            System.out.println("I know node express mongo");
        }
    }
    no usages
    public static void main(String[] args) {
        Backend b1 = new Backend();
        b1.skill();
    }
}
```

super(By constructor)

```java
public class vechile {
    1 usage  1 inheritor
    static class car{

        1 usage
        car(){
            System.out.println("car is there");
        }
        no usages
        car(String type){
            System.out.println("car is running "+type);
        }
//          String c = "car is running";
    }
    2 usages
    static class Bike extends car{
        1 usage
        Bike(){
            super();
            System.out.println("Bike is running");
        }
    }
```

super- By variable

```
no usages
public class sup {
    1 usage   1 inheritor
    static class A{
        1 usage
        String color = "black";
    }
    2 usages
    static class B extends A{
        1 usage
        void colorful(){
            System.out.println(super.color);
        }
        1 usage
        String color = "white";
    }

    no usages
    public static void main(String[] args) {
        B b1 = new B();
        System.out.println(b1.color);
        b1.colorful();
```

Hierarchical Inheritence