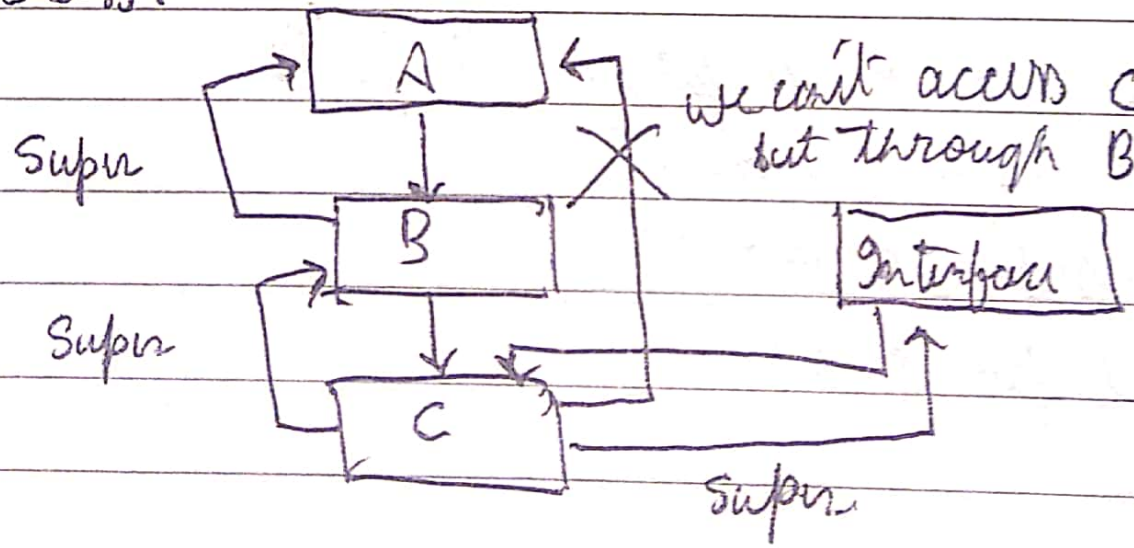
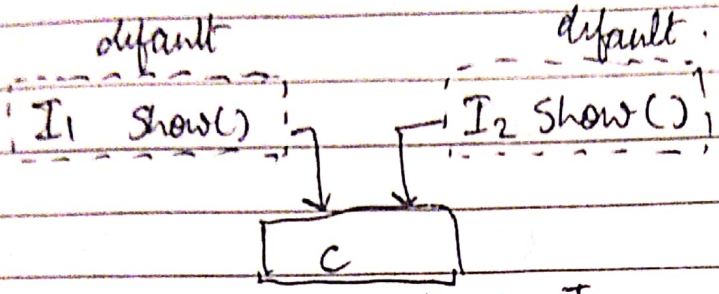
☆ Super Keyword:

with super we can access imdiate parent members which are overrided by childclass.



ambiguity problem :



(1) if show() is abstract then override in C and use.
(2) if show() an default then call them using super keyword.

```
I1.super.show()
I2.super.show()
```
} In this manner.

```
interface di1 {
    default void show(){
        Sop("show of di1");
    }
}
interface di2 {
    default void show(){
        Sop("show of di2");
    }
}
class superInterfaceDemo implements i1, i2 {
    Public void show(){
        di1.super.show();
        di2.super.show();
    }
```

Whenever child class constructor is called Parent class constructor is called

```
Public static void main (String [] args) {
    SuperInterfaceDemo obj = new SuperInterfaceDemo();
    obj.Show();
  }
}
```

✱ Super keyword override member:

```
class P {
    Protected int a;
    P() {
      a = 10;
    }
}
class C extends P {
    Protect int a;
    C() {
      a = 15
    }
  ● void Show() {
    S.O.P("Child class a = " + a);
    S.O.P("Parent class a = " + super.a);
  }
  Public static void main(String[] args) {
      C obj = new C();
      obj.Show();
  }
}
```

(If there is only parameterized and no default constructor you need to call it with super in child.)
(So always make default constructor of parent)

this is done automatically by compiler.

super()

```
class P1 {
    Public void display(){
        s.o.p("display of P1");
    }
}
class C1 extends P1 {
    public void display () {
        s.o.p("display of C1");
    }
    Void show() {
        display();
        super.display();
    }
    Public static void main(String[] args){
        C obj = new C();
        obj.show();
    }
}
```

☆ Super Keyword - Private member in child class:

```
class P3 {
    Private int l, w;
    P3() {
        l = 10;
        w = 15;
    }
    Public void show() {
```

```
    S.O.P(l*w);
  }
}
class C3 extends P3 {
    int l, w;
    C3() {
        l = 5;
        w = 7;
    }
    Public static void main(String[] args) {
        C3 obj = new C3();
        obj.show();
    }
}
```

- This will generate output 150 not 35 as show is binded with private member l, w of P3 class no P3() and show() one. encapsulated together hence, output 150.

```
class P4 {
    Private int l, w;
    P4(int a, int b) {
        l = a;
        w = b;
    Public Void area() {
        S.O.P(l*w);
}
class C4 extends P4 {
    int l, w;
```

explicit call of Parent constructor.

```
Cy (int a, int b) {
    super(a, b);
    l = a;
    w = b;
}
public static void main(String[] args){
    Cy obj = new Cy(10, 7);
    obj.area();
}
}
```

★ instanceof operator:

```
S.O.P (obj instanceof C3);
```

returns false ← if obj not object of C3.

→ returns true if obj is object of C3

★ Command line argument:

```
class cmdDemo {
    public static void main(String[] args){
        for(int i=0; i<arg.length(); i++){
            S.O.P(args[i]);
        }
    }
```

D:\ cou Java> Javac cmdDemo.Java

D:\ cou Java> Java cmdDemo Anil Sunil Ajay

O/P: Anil
Sunil
Ajay