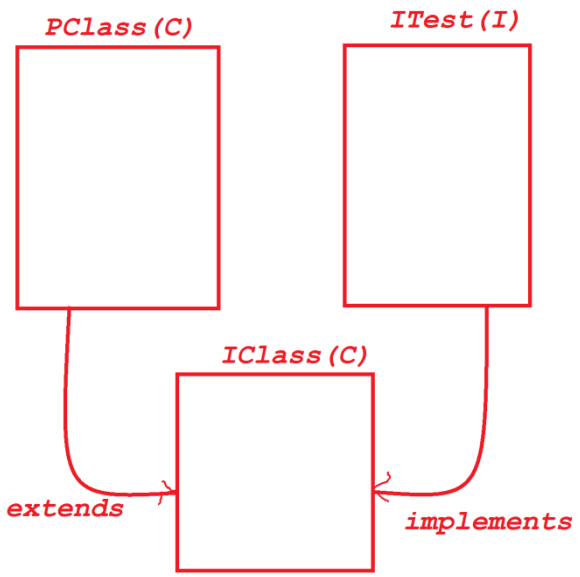
Dt : 26/8/2023

Model-2 : Extracting features from one class and any number of Interfaces into a class.

(Class extending from one class and implementing from any number of Interfaces)

Diagram:



Ex:

ProjectName : MultipleInheritance_App2

packages,

p1 : ITest.java

```
package p1;
public interface ITest {
   public abstract void m2(int b);
}
```

p1 : PClass.java

```
package p1;
public class PClass {
  public void m1(int a) {
       System.out.println("====m1(a)====");
       System.out.println("The value a:"+a);
  }
}
```

p1 : IClass.java

```
package p1;
public class IClass extends PClass implements ITest{
    public void m2(int b) {
     System.out.println("====m2(b)====");
     System.out.println("The value b:"+b);
     }
}
```

p2 : DemoMultipleInheritance2.java(MainClass)

```
package p2;
import p1.*;
public class DemoMultipleInheritance2 {
    public static void main(String[] args) {
       IClass ob = new IClass();
       ob.m1(11);
       ob.m2(23);
    }
}
```

o/p:

====m1(a)====

The value a:11

====m2(b)====

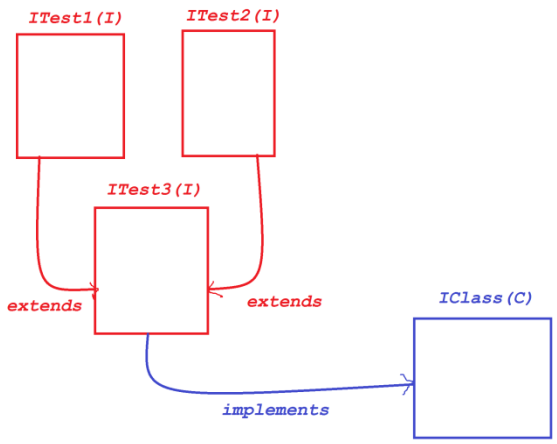
The value b:23

==========

Model-3 : Extracting features from more than one interface into a Interface

(Interface extending from more than one Interface)

Diagram:



Ex:

ProjectName : MultipleInheritance_App3

packages,

p1 : ITest1.java

```
package p1;
public interface ITest1 {
  public abstract void m1(int x);
}
```

p1 : ITest2.java

```
package p1;
public interface ITest2 {
    public abstract void m2(int y);
}
```

p1 : ITest3.java

```
package p1;
public interface ITest3 extends ITest1,ITest2{
    public abstract void m3(int z);
}
```

p1 : IClass.java

```
package p1;
public class IClass implements ITest3{
    public void m1(int x) {
    System.out.println("====m1(x)====");
    System.out.println("x:"+x);
    }
    public void m2(int y) {
        System.out.println("====m2(y)====");
        System.out.println("y:"+y);
        }
    public void m3(int z) {
        System.out.println("====m3(z)====");
        System.out.println("z:"+z);
        }
}
```

p2 : DemoMultipleInheritance3.java(MainClass)

```
package p2;
import p1.*;
public class DemoMultipleInheritance3 {
    public static void main(String[] args) {
```

```
        IClass ob = new IClass();
        ob.m1(1);
        ob.m2(2);
        ob.m3(3);
    }
}
```

o/p:

o/p:

====m1(x)====

x:1

====m2(y)====

y:2

====m3(z)====

z:3

==

****imp***

Generalization Process:

=>The process in which PClass holding the reference of CClass Object is known as Generalization process.

=>In generalization process the object will hold all the members of PClass and only Overriding members from the CClass.

syntax:

PClass ob = (PClass)new CClass();

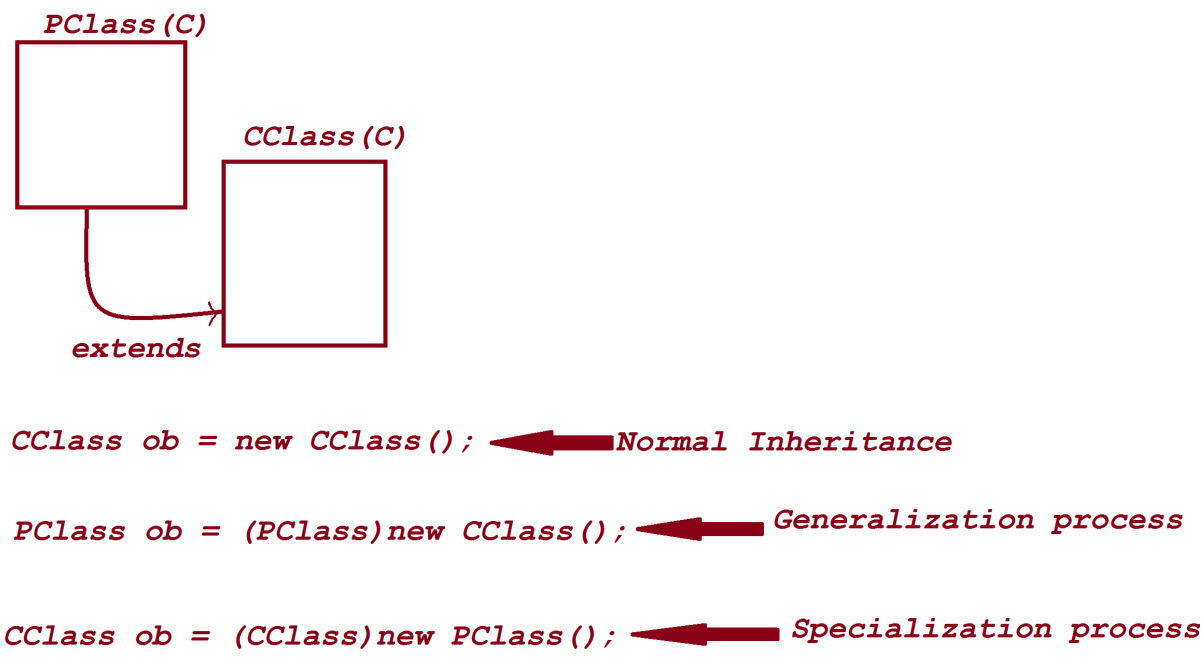
=>we can also perform Generalization process using Interfaces and AbstractClasses.

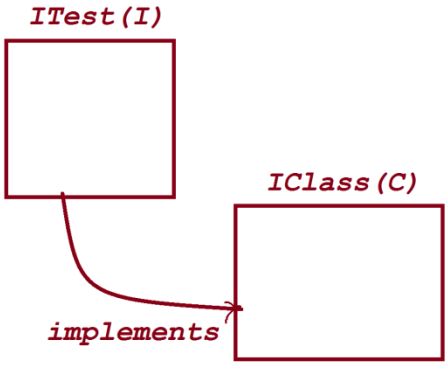
syntax:

ITest ob = (ITest)new IClass();

AClass ob = (AClass)new EClass();

Diagram:






```
IClass ob = new IClass();
```
Normal Inheritance

```
ITest ob = (ITest)new IClass();
```
Generalization process

~~`IClass ob = (IClass)new ITest();`~~
Specialization Process
