***Dt : 14/10/2022***

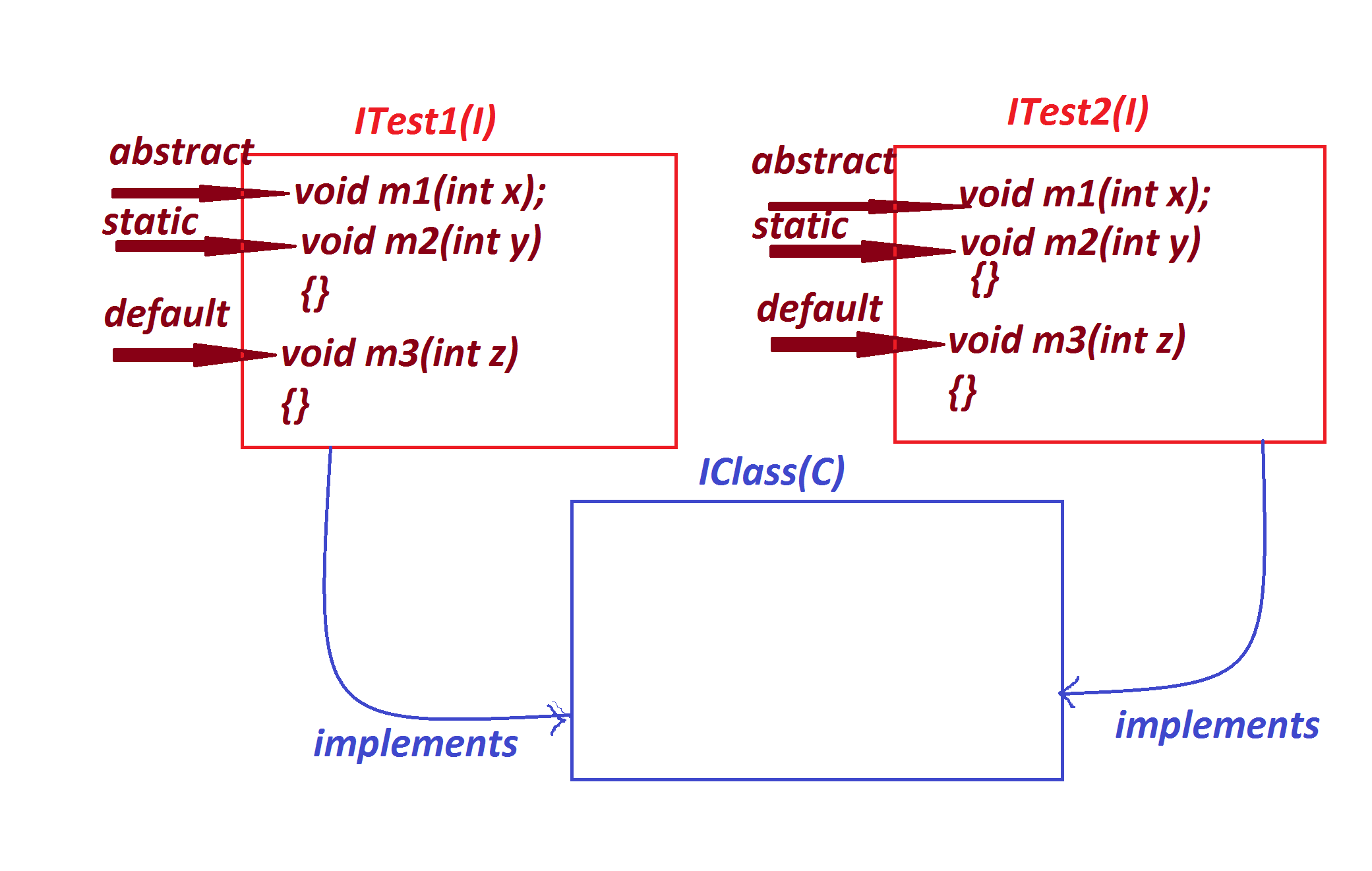
***\*imp***

***Multiple Inheritance models using Interfaces:***

***Model-1 : Extracting the features from more than one interface into a class***

***(Class implementing from more than one interface)***

***Diagram:***

******

***Ex:***

***ITest1.java***

***package test;***

***public interface ITest1 {***

***public abstract void m1(int x);***

***public static void m2(int y) {***

***System.out.println("===ITest1 static m2(y)====");***

***System.out.println("The value y:"+y);***

***}***

***public default void m3(int z,int p) {***

***System.out.println("===ITest1 default m3(z)====");***

***System.out.println("The value z:"+z);***

***this.m4(p);***

***}***

***private void m4(int p) {***

***System.out.println("====ITest1 private m4(p)===");***

***System.out.println("The value p:"+p);***

***}***

***}***

***ITest2.java***

***package test;***

***public interface ITest2 {***

***public abstract void m1(int x);***

***public static void m2(int y) {***

***System.out.println("===ITest2 static m2(y)====");***

***System.out.println("The value y:"+y);***

***}***

***public default void m33(int z,int p) {***

***System.out.println("===ITest2 default m33(z)====");***

***System.out.println("The value z:"+z);***

***this.m4(p);***

***}***

***private void m4(int p) {***

***System.out.println("====ITest2 private m4(p)===");***

***System.out.println("The value p:"+p);***

***}***

***}***

***IClass.java***

***package test;***

***public class IClass implements ITest1,ITest2{***

***public void m1(int x) {***

***System.out.println("====method m1(x)====");***

***System.out.println("The value x:"+x);***

***}***

***}***

***DemoInterface6.java***

***package maccess;***

***import test.\*;***

***public class DemoInterface6 {***

***public static void main(String[] args) {***

***IClass ob = new IClass();***

***ob.m1(123);***

***ITest1.m2(124);***

***ITest2.m2(125);***

***ob.m3(126,111);***

***ob.m33(127,222);***

***}***

***}***

***o/p:***

***====method m1(x)====***

***The value x:123***

***===ITest1 static m2(y)====***

***The value y:124***

***===ITest2 static m2(y)====***

***The value y:125***

***===ITest1 default m3(z)====***

***The value z:126***

***====ITest1 private m4(p)===***

***The value p:111***

***===ITest2 default m33(z)====***

***The value z:127***

***====ITest2 private m4(p)===***

***The value p:222***

***--------------------------------------------------------------***

***Note:***

***=>when we have same default concrete methods in Multiple-Inheritance***

***process then the ambiguity will be raised at compilation stage.***

***-------------------------------------------------------------***

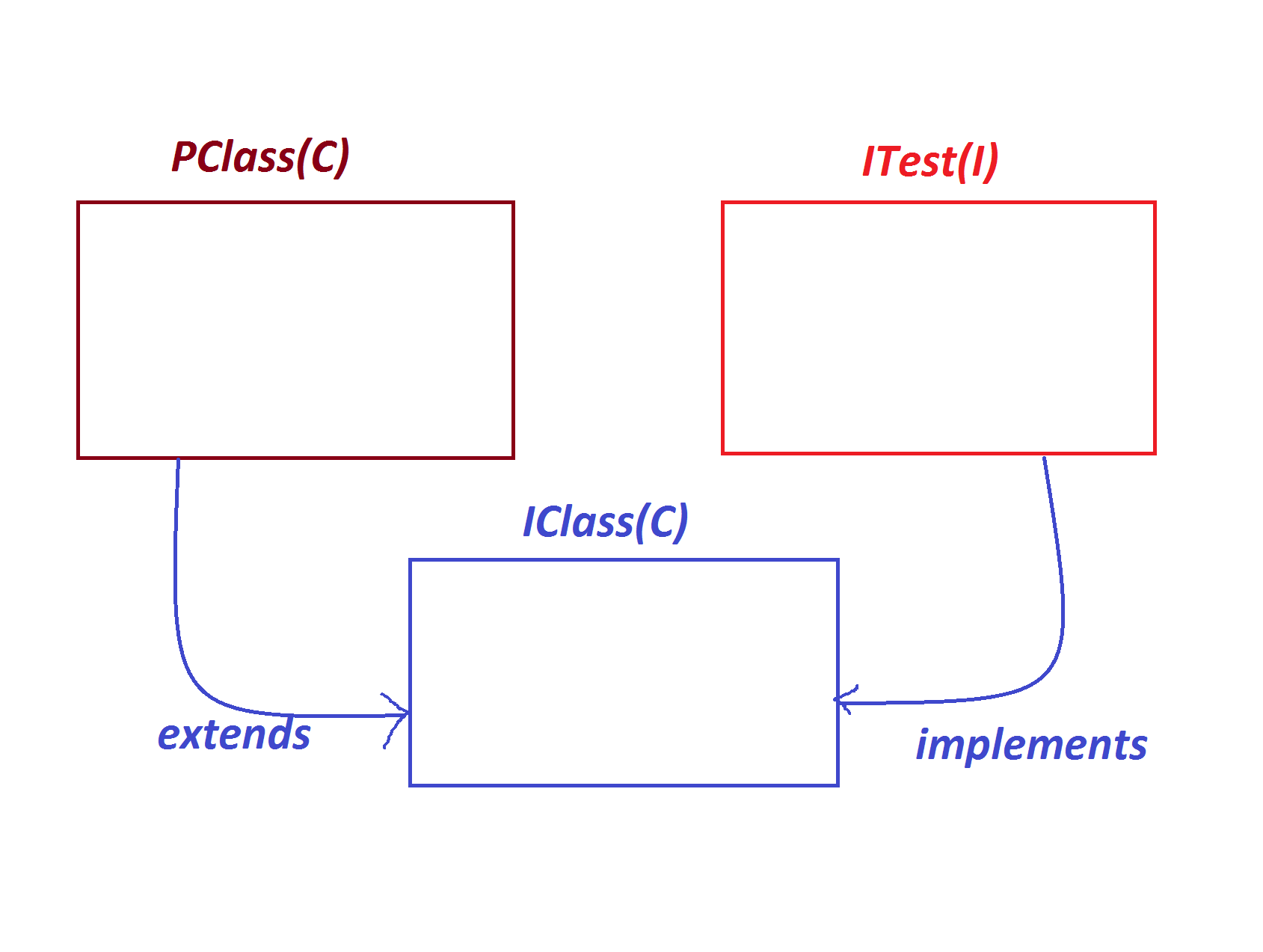
***Model-2 : Extracting features from one class and any number of Interfaces***

***into a Class.***

***(Class extending from one class and implementing from any number of***

***Interfaces)***

***Diagram:***

******

***Ex:***

***PClass.java***

***package test;***

***public class PClass {***

***public void m1(int x) {***

***System.out.println("====PClass m1(x)===");***

***System.out.println("The value x:"+x);***

***}***

***}***

***ITest.java***

***package test;***

***public interface ITest {***

***public abstract void m2(int y);***

***}***

***IClass.java***

***package test;***

***public class IClass extends PClass implements ITest{***

***public void m2(int y) {***

***System.out.println("====ITest m2(y)====");***

***System.out.println("The value y:"+y);***

***}***

***}***

***DemoInterface7.java(MainClass)***

***package maccess;***

***import test.\*;***

***public class DemoInterface7 {***

***public static void main(String[] args) {***

***IClass ob = new IClass();***

***ob.m1(12);***

***ob.m2(13);***

***}***

***}***

***o/p:====PClass m1(x)===***

***The value x:12***

***====ITest m2(y)====***

***The value y:13***

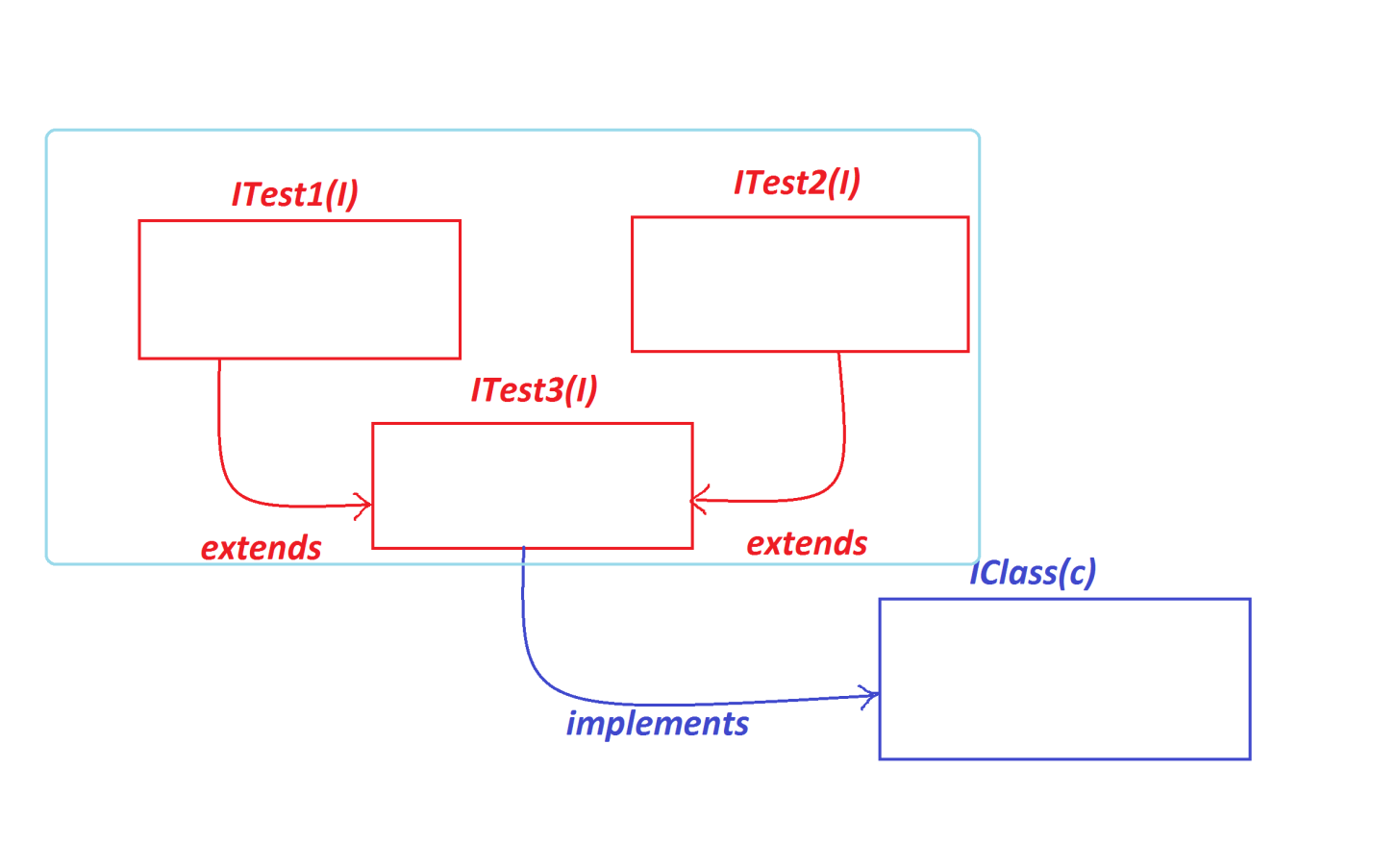
***------------------------------------------------------------------***

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

***Interface***

***(Interface extending from more than one Interfaces)***

***Diagram:***

******

***Ex:***

***ITest1.java***

***package test;***

***public interface ITest1 {***

***public abstract void m1(int x);***

***}***

***ITest2.java***

***package test;***

***public interface ITest2 {***

***public abstract void m2(int y);***

***}***

***ITest3.java***

***package test;***

***public interface ITest3 extends ITest1,ITest2{***

***public abstract void m3(int z);***

***}***

***IClass.java***

***package test;***

***public class IClass implements ITest3{***

***public void m1(int x) {***

***System.out.println("x:"+x);***

***}***

***public void m2(int y) {***

***System.out.println("y:"+y);***

***}***

***public void m3(int z) {***

***System.out.println("z:"+z);***

***}***

***}***

***DemoInterface8.java(MainClass)***

***package maccess;***

***import test.\*;***

***public class DemoInterface8 {***

***public static void main(String[] args) {***

***IClass ob = new IClass();***

***ob.m1(11);***

***ob.m2(12);***

***ob.m3(13);***

***}***

***}***

***o/p:***

***x:11***

***y:12***

***z:13***

***========================================================***

***Dt : 15/10/2022***

***Generalization process using Interfaces:***

***=>we can also perform Generalization process using interfaces and we use***

***the following syntax:***

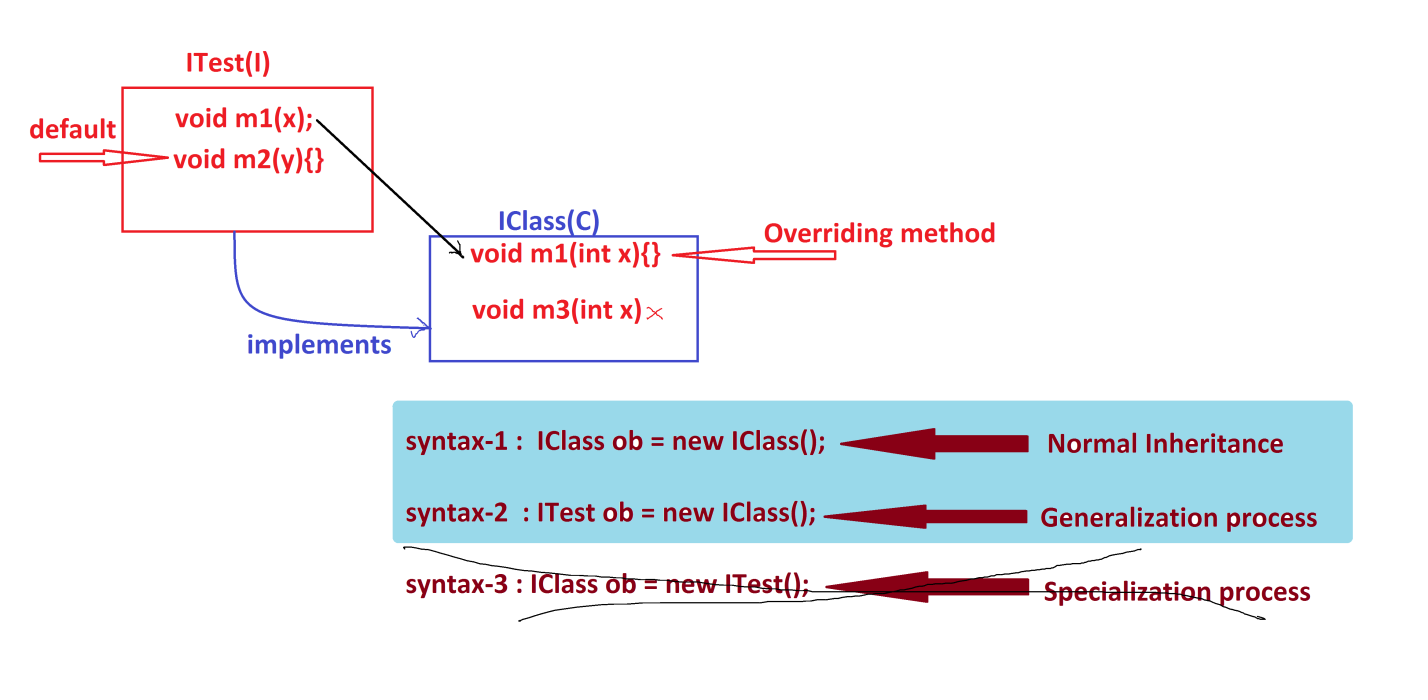
***ITest ob = new IClass();***

***=>In this Generalization process using Interfaces,one object created***

***holding all the members of Interface and only Overriding members from the***

***implementation class is known as Generalization process.***

***Diagram:***

******

***Ex:***

***ITest.java***

***package test;***

***public interface ITest {***

***public abstract void m1(int x);***

***public default void m2(int y) {***

***System.out.println("====default m2(y)====");***

***System.out.println("The value y:"+y);***

***}***

***}***

***IClass.java***

***package test;***

***public class IClass implements ITest{***

***public void m1(int x) {***

***System.out.println("====m1(int x)===");***

***System.out.println("The value x:"+x);***

***}***

***public void m3(int z) {***

***System.out.println("====m3(z)====");***

***System.out.println("The value z:"+z);***

***}***

***}***

***DemoInterface9.java(MainClass)***

***package maccess;***

***import test.\*;***

***public class DemoInterface9 {***

***public static void main(String[] args) {***

***System.out.println("\*\*\*\*Normal Inheritance\*\*\*\*");***

***IClass ob1 = new IClass();//Normal Inheritance***

***ob1.m1(11);***

***ob1.m2(12);***

***ob1.m3(13);***

***System.out.println("\*\*\*\*Generalization process\*\*\*\*");***

***ITest ob2 = (ITest)new IClass();***

***ob2.m1(111);***

***ob2.m2(222);***

***//ob.m3(333);//Error***

***//IClass ob3 = (IClass)new ITest();//Specialization Error***

***}***

***}***

***o/p:***

***\*\*\*\*Normal Inheritance\*\*\*\****

***====m1(int x)===***

***The value x:11***

***====default m2(y)====***

***The value y:12***

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

***The value z:13***

***\*\*\*\*Generalization process\*\*\*\****

***====m1(int x)===***

***The value x:111***

***====default m2(y)====***

***The value y:222***

***==================================================================***

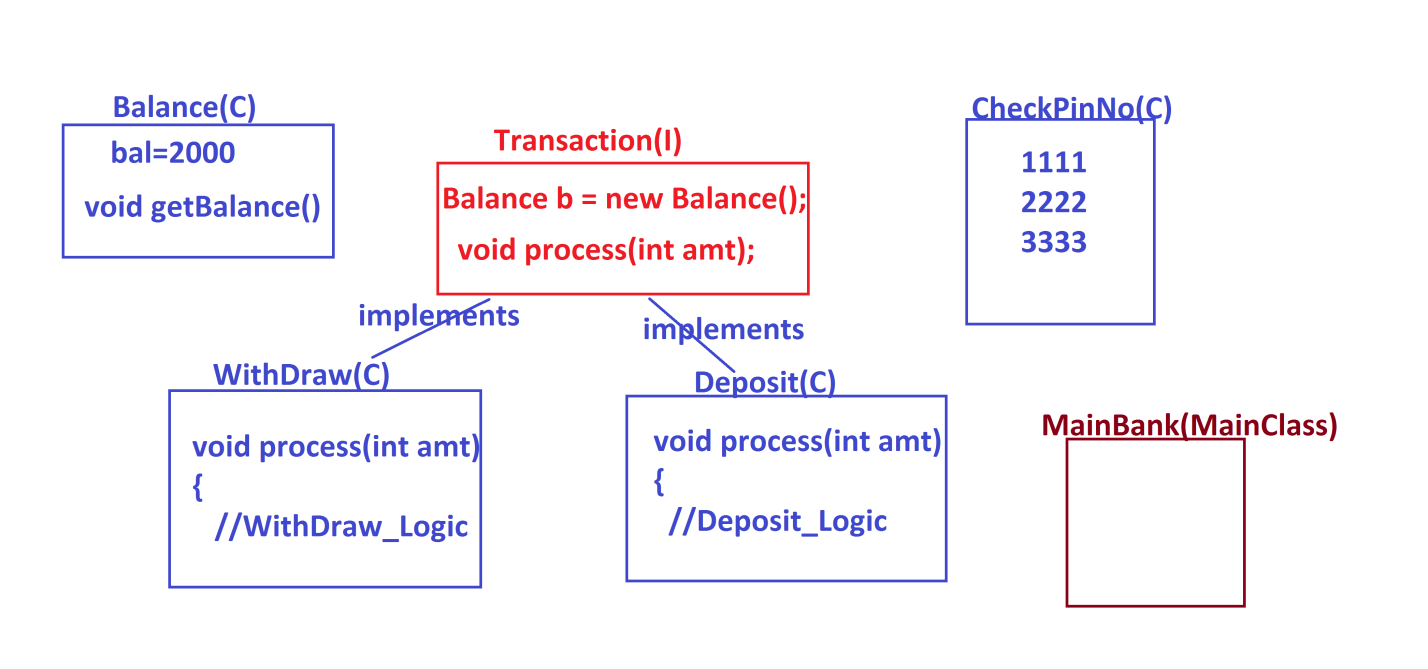
***Note:***

***=>we cannot perform Specialization process using Interfaces.***

***==============================================================***

***Assignment:***

***Construct BankTransaction application using the following Layout:***

******

***step-1 : read pinNo***

***=>pinNo must be in 1111 or 2222 or 3333,else "Invalid pinNo".***

***=>If pinNo entered wrongly for three times then display the msg as***

***"Transaction blocked".***

***step-2 : If the pinNo verified Successfully,then show the following choice:***

***1.WithDraw***

***2.Deposit***

***1.WithDraw:***

***=>Enter the amt***

***=>amt must be greater than Zero and multiples of 100,else "Invalid amt"***

***=>If the amt is validated Successfully,then create object for***

***"WithDraw" class and pass amt as parameter to "process()" method.***

***=>Perform WithDraw\_logic in process method,***

***=>If amt is less than balance then perform transaction,ele***

***display msg as "Insufficient fund"***

***o/p:***

***Amt withdrawn :***

***Balance amt :***

***Transaction Successfull***

***2.Deposit:***

***=>Enter the amt***

***=>amt must be greater than Zero and multiples of 100,else "Invalid amt"***

***=>If the amt is validated Successfully,then create object for***

***"Deposit" class and pass amt as parameter to "process()" method.***

***=>Perform Deposit\_logic in process method,***

***o/p:***

***Amt Deposited :***

***Balance amt :***

***Transaction Successfull***

***=======================================================================***

***\*imp***

***Abstract Classes in Java:***

***=>The classes which are declared with "abstract" keyword are known as***

***abstract classes.***

***=>Abstract Classes will hold Variables,Concrete methods,abstract methods,***

***Blocks,Constructors and features.***

***=>we must use "abstract" keyword to declare abstract methods in abstract***

***classes.***

***=>we cannot instantiate abstract classes,which means we cannot create***

***object for abstract classes.***

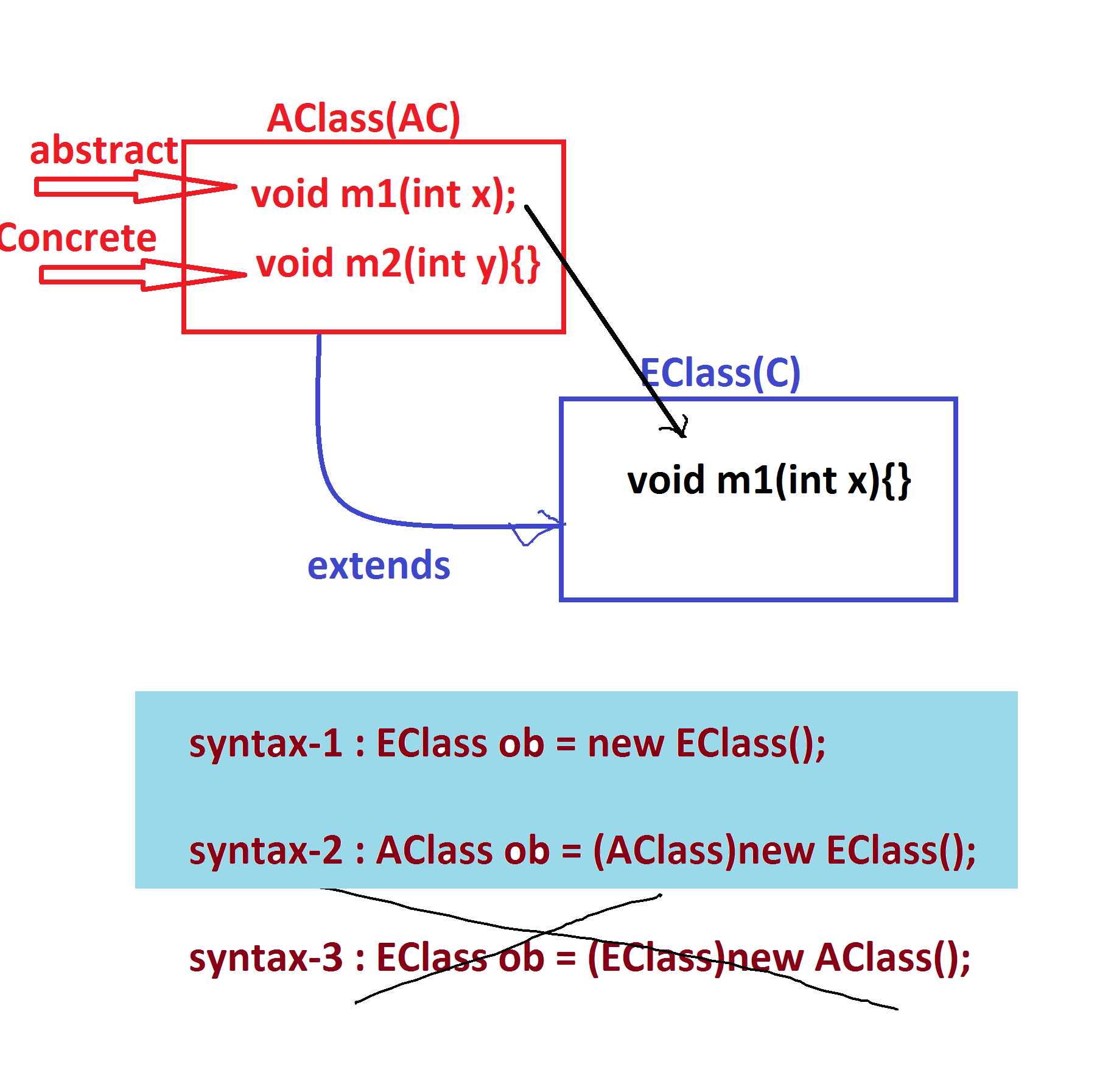
***=>These abstract classes must be extended to classes and the classes are***

***known as "extention classes" or "implemented classes".***

***=>We can also perform Generalization process using Abstract classes,but***

***we cannot perform Specialization process using AbstractClasses.***

***Diagram:***

******

***Ex:***

***AClass.java***

***package test;***

***public abstract class AClass {***

***public abstract void m1(int x);***

***public void m2(int y) {***

***System.out.println("=====Concrete m2(y)=====");***

***System.out.println("The value y:"+y);***

***}***

***}***

***EClass.java***

***package test;***

***public class EClass extends AClass{***

***public void m1(int x) {***

***System.out.println("====abstract m1(x)====");***

***System.out.println("The value x:"+x);***

***}***

***}***

***DemoAbstractClass.java(MainClass)***

***package maccess;***

***import test.\*;***

***public class DemoAbstractClass1 {***

***public static void main(String[] args) {***

***System.out.println("\*\*\*\*Normal Inheritance\*\*\*\*");***

***EClass ob1 = new EClass();***

***ob1.m1(11);***

***ob1.m2(12);***

***System.out.println("\*\*\*\*Generalization process\*\*\*\*");***

***AClass ob2 = (AClass)new EClass();***

***ob2.m1(11);***

***ob2.m2(12);***

***// EClass ob3 = (EClass)new AClass();//Error***

***}***

***}***

***o/p:***

***\*\*\*\*Normal Inheritance\*\*\*\****

***====abstract m1(x)====***

***The value x:11***

***=====Concrete m2(y)=====***

***The value y:12***

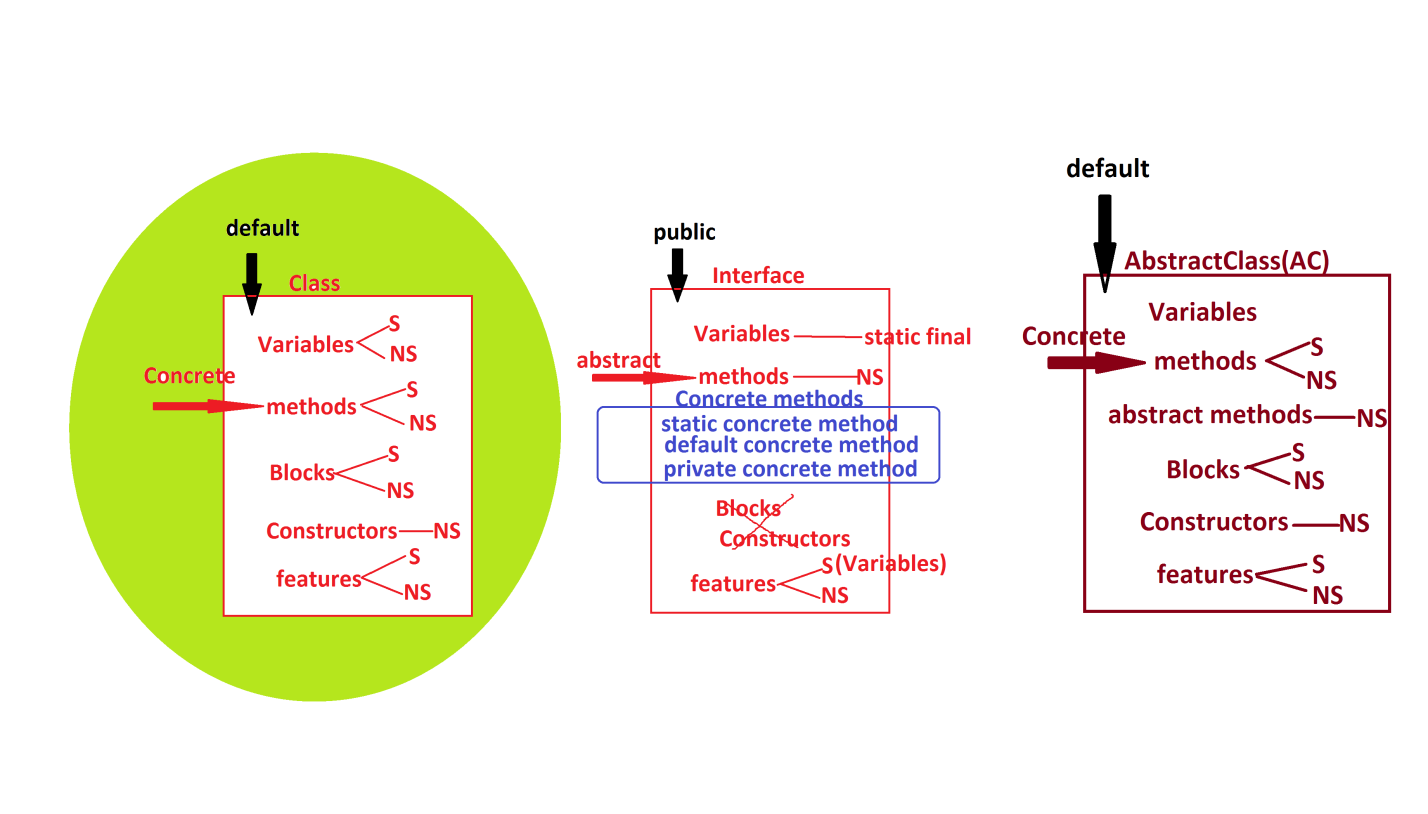
***\*\*\*\*Generalization process\*\*\*\****

***====abstract m1(x)====***

***The value x:11***

***=====Concrete m2(y)=====***

***The value y:12***

******