***Dt : 12/10/2022***

***faq:***

***wt is the diff b/w***

***(i)Implemented methods***

***(ii)Non-Implemented methods***

***(i)Implemented methods:***

***=>The methods which are taken from the interface and constructed***

***body are known as Implemented methods.***

***(ii)Non-Implemented methods:***

***=>The methods which are not taken from the interface are known***

***as Non-Implemented methods.***

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

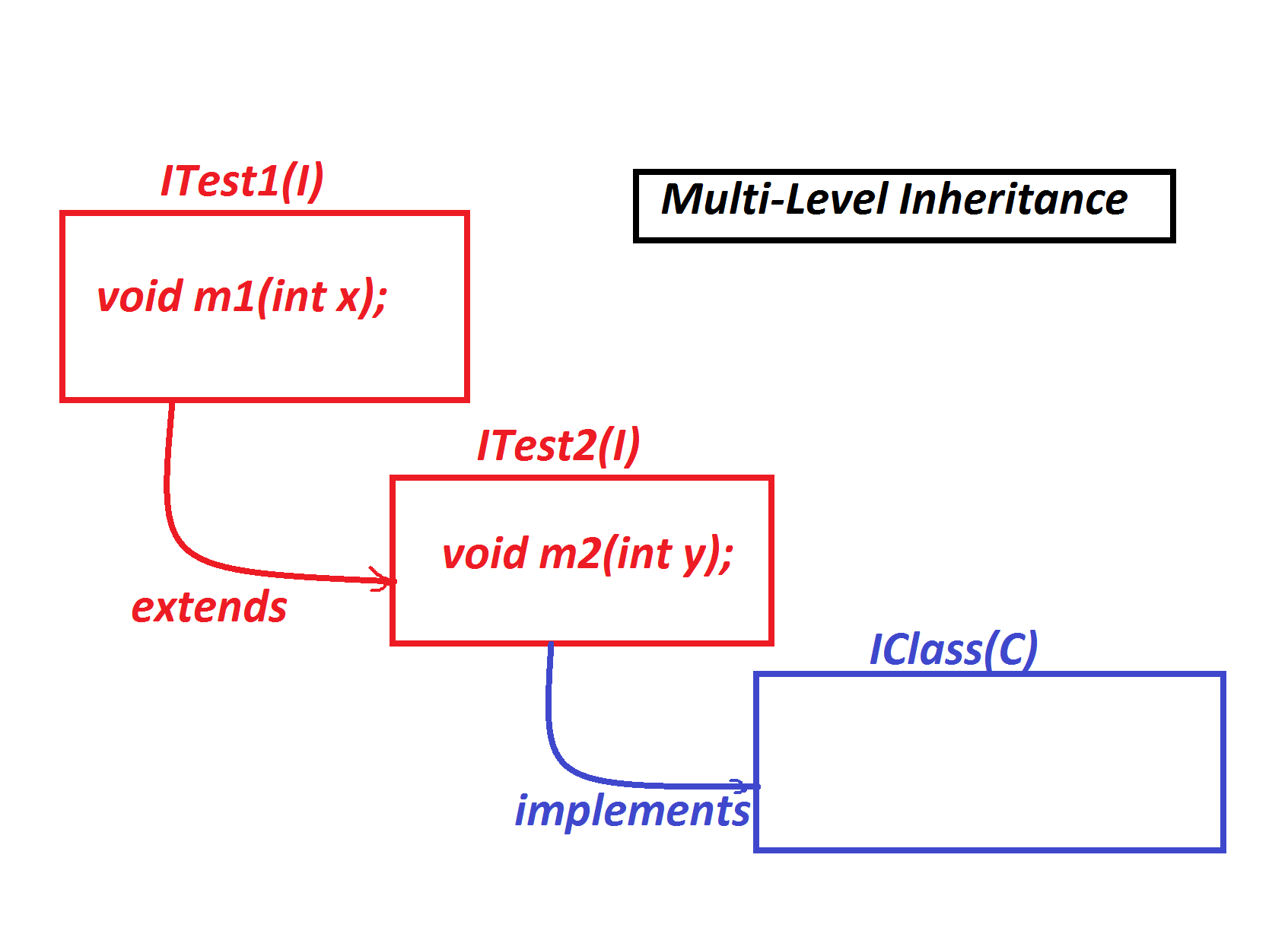
***Rule-12 : Interfaces cannot be declared with Blocks and***

***Constructors***

***Rule-13 : Interfaces can extract the features from another***

***Interfaces using "extends" keyword***

***Diagram:***

******

***Ex:***

***ITest1.java***

***package test;***

***public interface ITest1 {***

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

***}***

***ITest2.java***

***package test;***

***public interface ITest2 extends ITest1{***

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

***}***

***IClass.java***

***package test;***

***public class IClass implements ITest2{***

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

***System.out.println("===ITest1 m1(x)===");***

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

***}***

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

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

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

***}***

***}***

***DemoInterface2.java(MainClass)***

***package maccess;***

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

***public class DemoInheritance2 {***

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

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

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

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

***}***

***}***

***o/p:***

***===ITest1 m1(x)===***

***The value x:11***

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

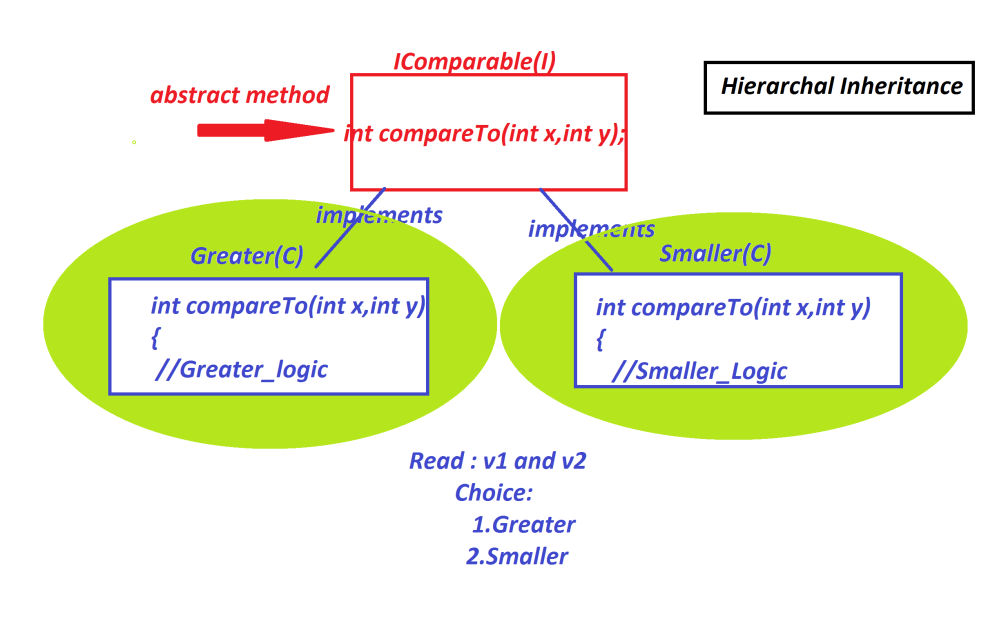
***The value y:12***

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

***Rule-14 : Interface can be implemented to any number of***

***Implementation classes.***

***Diagram:***

******

***Ex:***

***IComparable.java***

***package test;***

***public interface IComparable {***

***public abstract int compare(int x,int y);***

***}***

***GreaterValue.java***

***package test;***

***public class GreaterValue implements IComparable{***

***public int compare(int x,int y) {***

***if(x>y) return x;***

***else return y;***

***}***

***}***

***SmallerValue.java***

***package test;***

***public class SmallerValue implements IComparable{***

***public int compare(int x,int y) {***

***if(x<y) return x;***

***else return y;***

***}***

***}***

***DemoInterface3.java(MainClass)***

***package maccess;***

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

***import java.util.\*;***

***public class DemoInterface3 {***

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

***Scanner s = new Scanner(System.in);***

***System.out.println("Enter the value of x:");***

***int x = s.nextInt();***

***System.out.println("Enter the value of y:");***

***int y = s.nextInt();***

***System.out.println("====Choice====");***

***System.out.println("1.GreaterValue\n2.SmallerValue");***

***System.out.println("Enter the Choice:");***

***switch(s.nextInt())***

***{***

***case 1:***

***int r1 = new GreaterValue().compare(x, y);***

***System.out.println("GreaterValue:"+r1);***

***break;***

***case 2:***

***int r2 = new SmallerValue().compare(x, y);***

***System.out.println("SmallerValue:"+r2);***

***break;***

***default:***

***System.out.println("Invalid choice....");***

***}//end of switch***

***s.close();***

***}***

***}***

***o/p:***

***Enter the value of x:***

***12***

***Enter the value of y:***

***14***

***====Choice====***

***1.GreaterValue***

***2.SmallerValue***

***Enter the Choice:***

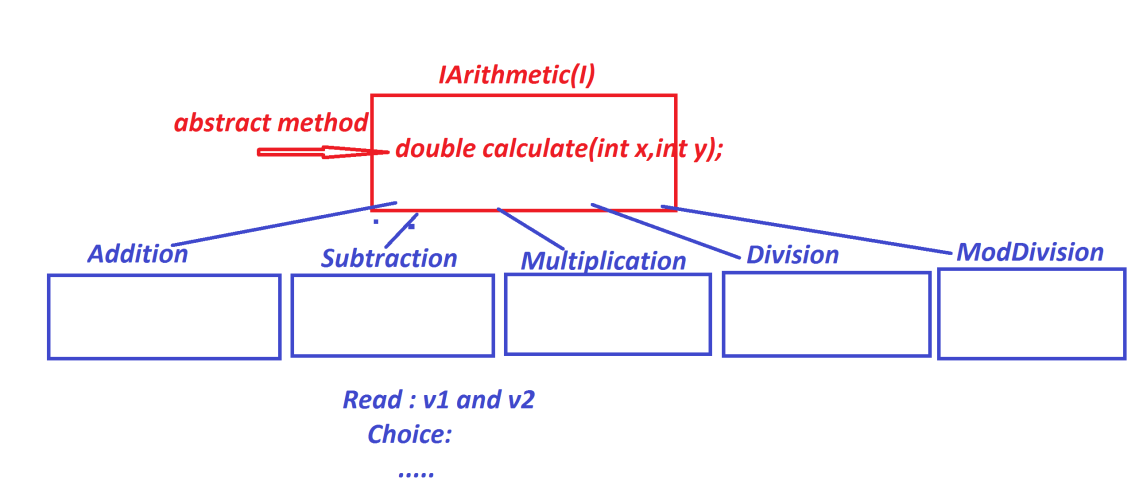
***1***

***GreaterValue:14***

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

***Assignment-1:***

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

******

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

***\*imp***

***Concrete methods in Interfaces:(Java8 - new feature)***

***=>From Java8 version onwards the interfaces can be declared with***

***Concrete methods.***

***=>The following are the Concrete methods can be declared in***

***Interfaces:***

***(a)static concrete methods(Java8)***

***(b)default concrete methods(Java8)***

***(c)private concrete methods(Java9)***

***(a)static concrete methods(Java8):***

***=>The concrete methods in interfaces which are declared with***

***"static" keyword are known as Static Concrete methods.***

***=>These static concrete methods will get the memory within the***

***interface while loading and accessed with interface\_name.***

***Coding Rule:***

***=>Static concrete methods of Interface are not available to***

***implementation classes,which means implementation classes cannot***

***access Static concrete methods.***

***Diagram:***

***Ex:***

***ITest.java***

***package test;***

***public interface ITest {***

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

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

***System.out.println("====static concrete 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(x)====");***

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

***}***

***}***

***DemoInterface4.java(MainClass)***

***package maccess;***

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

***public class DemoInterface4 {***

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

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

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

***ITest.m2(13);//Static method call***

***//IClass.m2(14);//Error***

***}***

***}***

***o/p:***

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

***The value x:12***

***====static concrete m2(y)=====***

***The value y:13***

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

***DT : 13/10/2022***

***(b)default concrete methods(Java8):***

***=>The concrete methods in interfaces which are declared with "default"***

***keyword are known as "default concrete methods".***

***Coding Rules:***

***=>These "default" concrete methods are NonStatic methods and available to***

***implementation classes and can be accessed with implementation class object***

***reference.***

***Note:***

***=>"default" keyword is not allowed in classes to declare programming***

***components.***

***=>To declare "default concrete methods" in interfaces "default" keyword is***

***manditory.***

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

***(c)private concrete methods(Java9)***

***=>The concrete methods in interfaces which are declared with "private"***

***keyword are known as private concrete methods.***

***=>These private concrete methods are categorized into two types:***

***(i)Static private concrete methods***

***(ii)Non-Static private Concrete methods***

***Coding Rule:***

***=>Private concrete methods are accessed by the Non-private methods of***

***Same interface.***

***Ex:***

***ITest.java***

***package test;***

***public interface ITest {***

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

***default void m2(int y,int a,int b) {***

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

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

***ITest.dis1(a);//static private method call***

***this.dis2(b);//Non static private method call***

***}***

***static private void dis1(int a) {***

***System.out.println("====dis1(a)====");***

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

***}***

***private void dis2(int b) {***

***System.out.println("====dis2(b)====");***

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

***}***

***}***

***IClass.java***

***package test;***

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

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

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

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

***}***

***}***

***DemoInterface5.java(MainClass)***

***package maccess;***

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

***public class DemoInterface5 {***

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

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

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

***ob.m2(12,111,222);//default concrete method\_call***

***//ITest.dis1(111);//Error***

***//ob.dis2(222);//Error***

***}***

***}***

***o/p:***

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

***The value x:11***

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

***The value y:12***

***====dis1(a)====***

***The value a:111***

***====dis2(b)====***

***The value b:222***

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