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

***Interfaces in Java:***

***=>Interface is a collection of Variables,abstract methods and***

***Concrete methods from Java8 version onwards.***

***(Upto Java7 version,Interface can hold Variables and abstract***

***methods,but cannot hold Concrete methods)***

***faq:***

***define abstract methods?***

***=>The methods which are declared without method\_body are known as***

***abstract methods***

***Structure of abstract methods:***

***return\_type method\_name(Para\_list);***

***faq:***

***define Concrete methods?***

***=>The methods which are declared with method\_body are known as***

***Concrete methods.***

***Structure of Concrete methods:***

***return\_type method\_name(Para\_list)***

***{***

***//method\_body***

***}***

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

***Coding rules of Interface:***

***Rule-1 : we use 'interface' keyword to declare interfaces.***

***syntax:***

***interface Interface\_name***

***{***

***//memebers***

***}***

***Rule-2 : The members which are declared within the interface are***

***automatically 'public'***

***Note:***

***=>The members which are declare within the class are automatically***

***"default"***

***Rule-3 : we can declare primitive datatype variables and***

***Non-primitive datatype variables in interface.***

***Rule-4 : The variables which are declared in interfaces are***

***are automatically 'static' and 'final' variables.***

***(i)static variables in interfaces will get the memory in interfaces***

***while interface loading and can access with Interface\_names.***

***(ii)final variables must be initialized with values and once***

***initialized cannot be modified.***

***(final variables are constant variables and Secured variables)***

***Dt : 11/10/2022***

***faq:***

***can we declare Non-Static Variables in interfaces?***

***=>No,we cannot declare NonStatic variables in interfaces,because***

***the variables are automatically static variables.***

***Rule-5 : The methods which are declared in interfaces are***

***automatically NonStatic abstract methods.***

***Note:***

***=>There is no concept of static abstract methods in Interfaces.***

***Rule-6 : Interfaces cannot be instantiated,which means we cannot***

***create object for Interfaces.***

***Rule-7 : Interfaces are implemented to classes using "implements"***

***keyword and the classes are known as implementation***

***classes.***

***syntax:***

***class ImplClass implements Interface***

***{***

***//members***

***}***

***Rule-8 : Implementation classes must construct body for abstract***

***methods of Interfaces.***

***Note:***

***=>Create object for implementation classes.***

***Rule-9 : Interfaces canbe declared with any number of abstract***

***methods without restriction.***

***Rule-10 : Implementation classes must construct body for all***

***abstract methods of interfaces.***

***Rule-11 : Implementation classes can also be declared with***

***Non-implemented methods.***

***EX:***

***IArithmetic.java***

***package test;***

***public interface IArithmetic {***

***int k=200;***

***double calculate(int x,int y);***

***void dis();***

***}***

***Addition.java***

***package test;***

***public class Addition implements IArithmetic{***

***public double calculate(int x,int y)//Implemented and Overriding method***

***{***

***return x+y;***

***}***

***public void dis() //Implemented and Overriding method***

***{***

***System.out.println("====method dis()====");***

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

***}***

***public void m(int p)//Non-Implemented method***

***{***

***System.out.println("====method m(p)=====");***

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

***}***

***}***

***DemoInterface1.java(MainClass)***

***package maccess;***

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

***public class DemoInterface1 {***

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

***System.out.println("Value k:"+IArithmetic.k);***

***//IArithmetic.k = 300;//Error***

***//IArithmetic ob = new IArithmetic();//Error***

***Addition ad = new Addition();***

***//implementation object of Interface***

***double r = ad.calculate(12, 13);***

***System.out.println("Sum:"+r);***

***ad.dis();***

***ad.m(123);***

***}***

***}***

***o/p:***

***Value k:200***

***Sum:25.0***

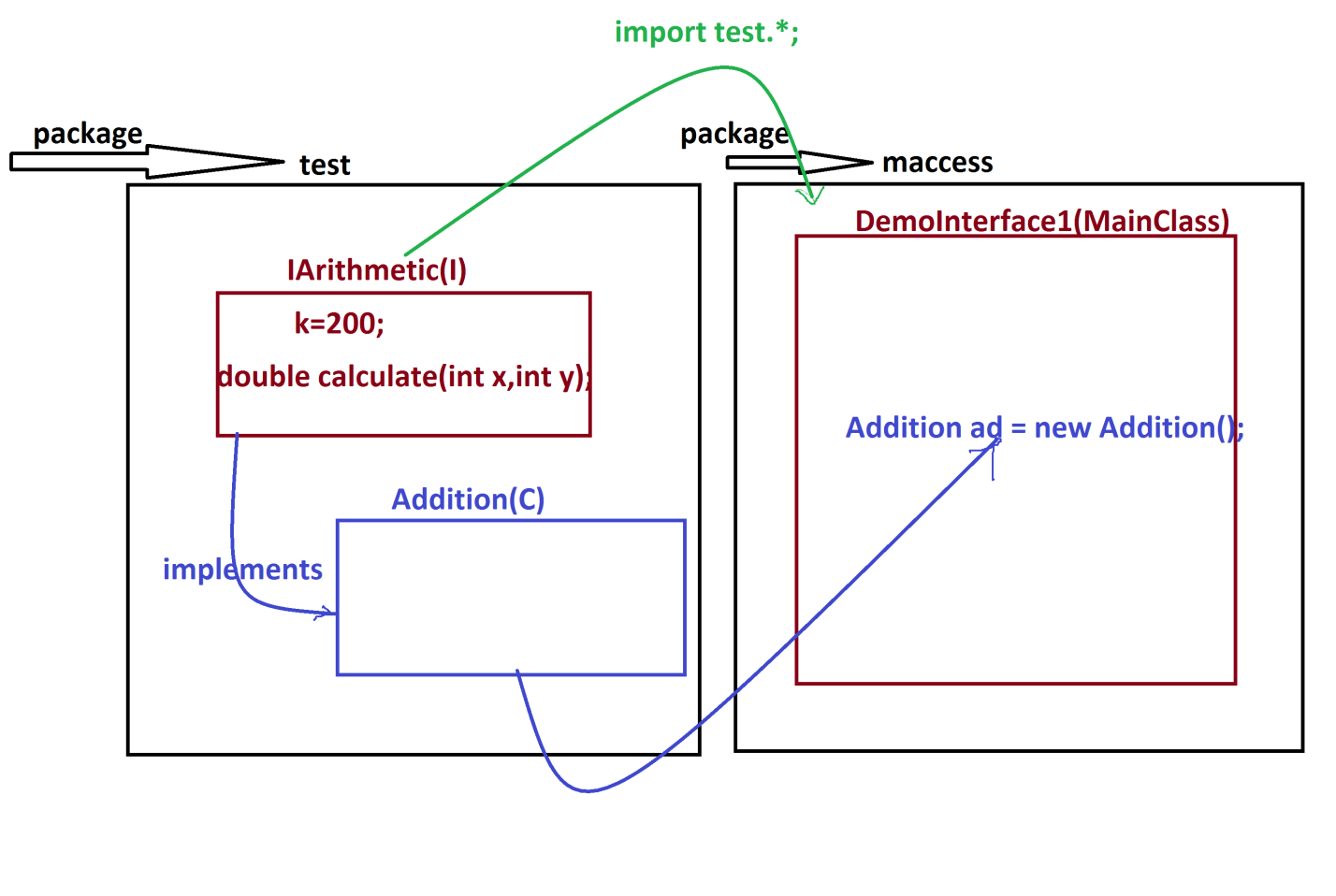
***====method dis()====***

***The value k:200***

***====method m(p)=====***

***The value p:123***

***Diagram:***

******

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