***Dt : 4/11/2022***

***\*imp***

***2.private:***

***=>The following are the private programming components:***

***(a)private variables***

***(b)private methods***

***(c)private Constructors***

***(d)private classes***

***=>There is no concept of private Blocks,Private Interfaces and private***

***Abstract Classes.***

***(a)private variables:***

***=>The variables which are declared with private keyword are known as***

***private variables.***

***Coding rule:***

***=>private variables are accessed by the NonPrivate methods of same class,***

***which private variables are available to the methods declared in the same***

***class.***

***Note:***

***=>In realtime private variables are used in Bean classes and POJO classes.***

***(POJO - Plain Old Java Object)***

***(b)private methods:***

***=>The methods which are declared with private keyword are known as***

***private methods.***

***Coding Rule :***

***=>private methods are accessed by the NonPrivate methods of same class.***

***\*imp***

***(c)private Constructors:***

***=>The constructors which are declared with private keyword are known as***

***private Contructors***

***Coding Rule:***

***=>Privtae Constructor is executed when the object is created inside the***

***same class where constructor is available,which means we cannot create object***

***outside the class.***

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

***faq:***

***define SingleTon classes?***

***=>The classes which generate only one object are known as SingleTon***

***Classes.***

***faq:***

***define SingleTon class design pattern?***

***=>The pattern which is used to construct SingleTon classes is known as***

***SingleTon class design pattern.***

***=>The following components are used in SingleTon class design pattern:***

***(i)private static reference variable***

***(ii)private Constructor***

***(iii)static method***

***(i)private static reference variable:***

***=>private static reference variable will hold the reference of object***

***created inside the class.***

***(ii)private Constructor:***

***=>private Constructor will restrict the object creation from externally.***

***(iii)static method:***

***=>static method is used to access the object reference outside the class***

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

***=>Based on Object creation process "SingleTon class design pattern" is***

***categorized into two types:***

***(1)Early Instantiation process***

***(2)Late Instantiation process***

***(1)Early Instantiation process:***

***=>In Early Instantiation process the object is created using "static"***

***block.***

***Ex:***

***PTest.java***

***package test;***

***public class PTest {***

***private int k=100;***

***private PTest() {}***

***private static PTest ob = null;***

***static***

***{***

***ob = new PTest();//Private\_Con\_Call***

***}***

***public static PTest getReference()***

***{***

***return ob;***

***}***

***private void dis()***

***{***

***System.out.println("\*\*\*\*private void dis()\*\*\*\*\*");***

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

***}***

***public void access()***

***{***

***System.out.println("\*\*\*\*public void access()\*\*\*\*");***

***System.out.println("private variable k:"+k);***

***this.dis();***

***}***

***}***

***DemoPoly1.java(MainClass)***

***package maccess;***

***import test.PTest;***

***public class DemoPoly1 {***

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

***PTest ob = PTest.getReference();//Accessing Object reference***

***//System.out.println("The value of k:"+ob.k);//Error***

***// ob.dis();//Error***

***ob.access();***

***}***

***}***

***o/p:***

***\*\*\*\*public void access()\*\*\*\****

***private variable k:100***

***\*\*\*\*private void dis()\*\*\*\*\****

***The value k:100***

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

***(2)Late Instantiation process:***

***=>In Late Instantiation process the object is created using method.***

***(Late Instantiation process is also known as Lazy Instantiation process)***

***Ex:***

***PTest2.java***

***package test;***

***public class PTest2{***

***private int k=100;***

***private PTest2() {}***

***private static PTest2 ob = null;***

***public static PTest2 getReference()***

***{***

***if(ob==null)***

***{***

***ob = new PTest2();//Private\_Con\_Call***

***}***

***return ob;***

***}***

***private void dis()***

***{***

***System.out.println("\*\*\*\*private void dis()\*\*\*\*\*");***

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

***}***

***public void access()***

***{***

***System.out.println("\*\*\*\*public void access()\*\*\*\*");***

***System.out.println("private variable k:"+k);***

***this.dis();***

***}***

***}***

***DemoPoly2.java(MainClass)***

***package maccess;***

***import test.PTest2;***

***public class DemoPoly2 {***

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

***PTest2 ob = PTest2.getReference();//Accessing Object reference***

***//System.out.println("The value of k:"+ob.k);//Error***

***// ob.dis();//Error***

***ob.access();***

***}***

***}***

***o/p:***

***\*\*\*\*public void access()\*\*\*\****

***private variable k:100***

***\*\*\*\*private void dis()\*\*\*\*\****

***The value k:100***

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

***(d)private classes:***

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

***Private classes***

***Coding Rule:***

***=>Private classes can be declared as only InnerClasses,which means the***

***OuterClasses cannot be private.***

***Note:***

***=>Private InnerClass object is created inside the NonPrivate method of***

***Same class.***

***Ex:***

***SubClass1.java***

***package test;***

***public class SubClass1 {***

***private class SubClass2{***

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

***System.out.println("\*\*\*\*private InnerClass m2(x)\*\*\*\*\*");***

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

***}***

***}//Private InnerClass***

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

***SubClass2 ob2 = new SubClass2();//Private InnerClass object***

***ob2.m2(x);***

***}***

***}//OuterClass***

***DemoPoly3.java(MainClass)***

***package maccess;***

***import test.SubClass1;***

***public class DemoPoly3 {***

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

***SubClass1 ob1 = new SubClass1();//OuterClass Object***

***ob1.access(123);***

***}***

***}***

***o/p:***

***\*\*\*\*private InnerClass m2(x)\*\*\*\*\****

***The value x:123***

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