***Dt : 22/9/2022***

***Note:***

***=>Generalization process is used to control ChildClass to have***

***only Overriding methods.***

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

***faq:***

***define Specialization process?***

***=>The process of Constructing ChildClass based on one feature***

***from the ParentClass is known as Specialization process.***

***=>we use the following syntax to perform Specialization process:***

***CClass ob = (CClass)new PClass();***

***Coding rule:***

***=>In Specialization process the PClass must be Pre-defined class***

***from the lib,if not raises 'java.lang.ClassCastException'***

***Note:***

***=>This Specialization process is also known as DownCasting process***

***or Narrowing process or Explicit TypeCasting process.***

***Ex:***

***Serialization and DeSerialization process***

***Cloning process***

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

***faq:***

***define Method Overloading process?***

***=>More than one method with same method name but differentiated***

***by their para\_list or para\_type is known as Method Overloading***

***process.***

***Note:***

***=>return\_type of methods are not included in Method Overloading***

***process.***

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

***Note:***

***=>we can perform Instance method Overloading process,Static***

***method Overloading process and Constructor Overloading process.***

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

***faq:***

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

***(i)super()***

***(ii)this()***

***(i)super():***

***=>super() is used to access the constructors from the PClass or***

***SuperClass.***

***(ii)this():***

***=>this() is used to access the constructors from the same class***

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

***faq:***

***define Constructor Chaining process or Constructor Interlinking***

***process?***

***=>The process of calling constructor from the constructor is known***

***as Constructor Chaining process.***

***Note:***

***=>super() is used to interlink CClass constructor to PClass***

***Constructor.***

***=>this() is used to interlink constructors from the same class***

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

***faq:***

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

***(i)this***

***(ii)super***

***(i)this:***

***=>'this' keyword is used to access the variables and methods***

***from the same class.***

***(ii)super:***

***=>'super' keyword is used to access the variables and methods***

***from the PClasss or SuperClass.***

***Note:***

***=>'this' and 'super' are pre-defined Non-static reference variables***

***and must be used only in Instance methods.***

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

***faq:***

***can we use 'super' and 'this' keywords to access static members?***

***=>Yes,we can use 'super' and 'this' keywords to access static***

***members,but these keywords must be used in Instance methods.***

***Note:***

***=>we cannot interlink static methods for execution.***

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

***Ex :***

***Display.java***

***package test;***

***public class Display {***

***public Display(int x) {***

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

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

***}***

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

***this(x);//Con\_Call\_from\_same\_Call***

***System.out.println("====Display(x,y)====");***

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

***}***

***public void m(int p) {***

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

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

***}***

***public void m(int p,int q) {***

***this.m(p);//method\_call\_from\_same\_class***

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

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

***}***

***public static void show(int a) {***

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

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

***}***

***public static void show(int a,int b) {***

***//this.show(a);//Compilation Error***

***System.out.println("====show(a,b)====");***

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

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

***}***

***public void access(int a,int b) {***

***this.show(a);***

***this.show(a, b);***

***}***

***}***

***DemoInheritance5.java(MainClass)***

***package maccess;***

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

***public class DemoInheritance5 {***

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

***Display d = new Display(12,13);***

***d.m(100,200);***

***d.access(12, 13);***

***}***

***}***

***o/p:***

***====Display(x)====***

***x:12***

***====Display(x,y)====***

***y:13***

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

***p:100***

***====m(p,q)====***

***q:200***

***====show(a)====***

***The value a:12***

***====show(a,b)====***

***The value a:12***

***The value b:13***

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

***Note:***

***=>Method Overloading process is also known Method Extension***

***=>Method Overriding process is also known as Method replacement.***

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