***Dt : 29/9/2022***

***Case-2 : Instance method Overloading process***

***=>More than one instance method differentiated by their paraList***

***or paraType is known as Instance method Overloading process.***

***Ex:***

***PClass.java***

***package test;***

***public class PClass {***

***public int a=200;***

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

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

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

***}***

***}***

***Display.java***

***package test;***

***public class Display extends PClass{***

***public int a=300;***

***public void m(int x,int y,int z) {***

***this.m(x, y);//method\_with\_2\_para\_from\_same\_class***

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

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

***}***

***public void m(int x,int y) {***

***super.m(x);//PClass\_method\_with\_1\_para***

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

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

***}***

***public void dis() {***

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

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

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

***}***

***}***

***DemoInheritance7.java(MainClass)***

***package maccess;***

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

***public class DemoInheritance7 {***

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

***Display ob = new Display();***

***ob.m(12,13, 14);//method\_with\_3\_para***

***ob.dis();***

***}***

***}***

***o/p:***

***====m(x)===***

***x:12***

***====m(x,y)====***

***y:13***

***====m(x,y,z)====***

***z:14***

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

***ParentClass value : 200***

***ChildClass value : 300***

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

***faq:***

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

***(i)super***

***(ii)this***

***(i)super:***

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

***from the PClass or SuperClass***

***(ii)this:***

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

***the Same class or Current Running class.***

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

***Case-3 : Static method Overloading process***

***=>More than one static method differentiated by their paraList or***

***ParaType is known as static method Overloading process.***

***Ex:***

***PClass.java***

***package test;***

***public class PClass {***

***public static void m(int x) {***

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

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

***}***

***}***

***CClass.java***

***package test;***

***public class CClass extends PClass{***

***public static void m(int x,int y,int z) {***

***//this.m(x,y);//Error***

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

***System.out.println("x:"+x+"\ny:"+y+"\nz:"+z);***

***}***

***public static void m(int x,int y) {***

***//super.m(x);//Error***

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

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

***}***

***public void access(int x,int y,int z) {***

***super.m(x);***

***this.m(x, y);***

***this.m(x, y, z);***

***}***

***}***

***DemoInheritance8.java(MainClass)***

***package maccess;***

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

***public class DemoInheritance8 {***

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

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

***ob.access(11, 12, 13);***

***}***

***}***

***o/p:***

***====m(x)=====***

***x:11***

***====m(x,y)=====***

***x:11***

***y:12***

***====m(x,y,z)=====***

***x:11***

***y:12***

***z:13***

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

***faq:***

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

***=>Yes,we can access static members using 'super' and 'this' keywords***

***but these keywords must be used in NonStatic methods,because 'super'***

***and 'this' keywords are NonStatic NonPrimitive datatype variables.***

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

***faq:***

***can we perform Method Overriding process for standard main() method?***

***=>No,we cannot perform Method Overriding process for standard main()***

***becuase standard main() method is static method.***

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

***faq:***

***can we perform Method Overloading process for standard main() method?***

***=>Yes,we can perform Method Overloading process for Standard main()***

***method,because static method Overloading is posible***

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

***faq:***

***can we pass parameters to Standard main() method?***

***=>Yes,we can pass parameters to standard main() method while***

***execution process,because the main() method call is available in***

***execution command.***

***syntax:***

***java Class\_name arg1 arg2 arg3 ...***

***Ex:***

***DemoInheritance9.java(MainClass)***

***package maccess;***

***public class DemoInheritance9 {***

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

***{***

***DemoInheritance9.main(123);***

***DemoInheritance9.main(12.34F);***

***System.out.println("====main(String[])=====");***

***for(String p : args)***

***{***

***System.out.println(p.toString());***

***}***

***}***

***public static void main(int k)***

***{***

***System.out.println("=====main(int)====");***

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

***}***

***public static void main(float z)***

***{***

***System.out.println("=====main(float)====");***

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

***}***

***}***

***o/p:***

***E:\Demo136>java DemoInheritance9 nit hyd thread task***

***=====main(int)====***

***The value k:123***

***=====main(float)====***

***The value z:12.34***

***====main(String[])=====***

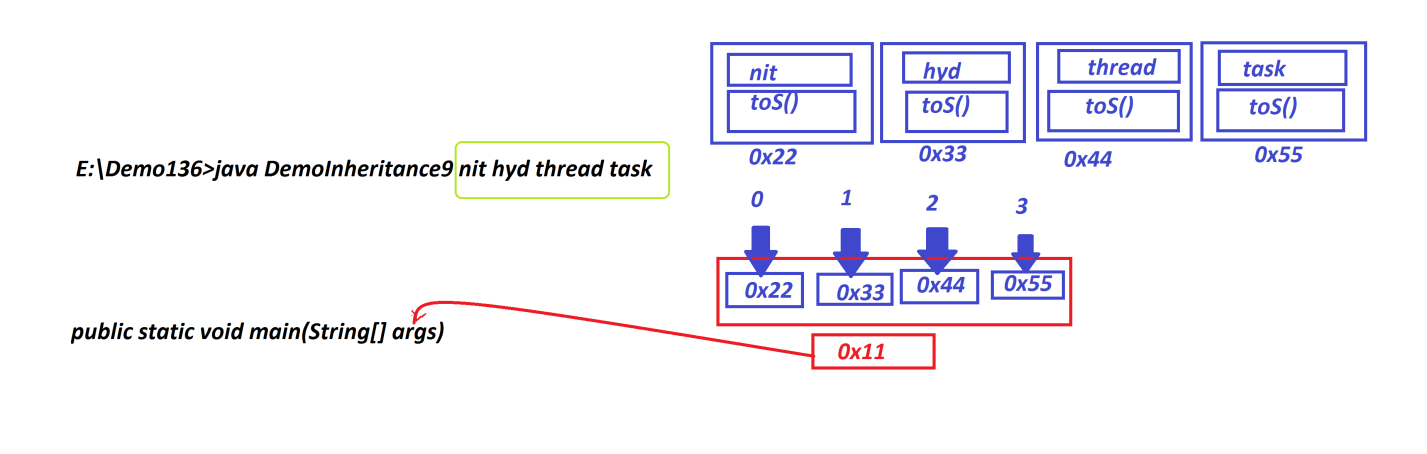
***nit***

***hyd***

***thread***

***task***

***Diagram:***

******

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

***faq:***

***define CommandLine argument program?***

***=>The program in which we pass arguments to standrad main() method***

***is known as CommandLine argument program.***

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