Dt : 1/10/2022

Note:

=>There is no concept of Constructor Overriding process,because the Constructors will have different names in PClass and CClass.

=>There is no concept of static method Overriding process,because the static members will get the memory within the class and available within the class.

=>Instance method Overriding is posible,because the Instance methods will get the memory within the object while object creation process.

Ex-program:

PClass.java

```
package test;
public class PClass {
    public PClass(int x)
    {
     System.out.println("===PClass(x)====");
     System.out.println("The value x:"+x);
    }
    public static void m1(int a)//memory in PClass
    {
     System.out.println("====PClass static m1(a)====");
     System.out.println("The value a:"+a);
    }
    public void m2(int p)//memory in Object
    {
     System.out.println("====PClass Instance m2(p)====");
     System.out.println("The value p:"+p);
    }
}
```

CClass.java

```
package test;
public class CClass extends PClass
```

```
{
     public CClass(int x)
     {
      super(x);//PClass_Con_Call
     }
     public static void m1(int a)//memory in CClass
     {
          System.out.println("====CClass static m1(a)====");
          System.out.println("The value a:"+a);
     }
     public void m2(int p)//memory in Object
     {
          System.out.println("====CClass Instance m2(p)====");
          System.out.println("The value p:"+p);
     }
}
```

DemoInheritance4.java(MainClass)

```
package maccess;
import test.*;
public class DemoInheritance4 {
     public static void main(String[] args) {
        CClass ob = new CClass(123);//CClass_Con_Call
        CClass.m1(124);
        ob.m2(125);
     }
}
```

o/p:

```
===PClass(x)====

The value x:123

====CClass static m1(a)====

The value a:124

====CClass Instance m2(p)====

The value p:125
```

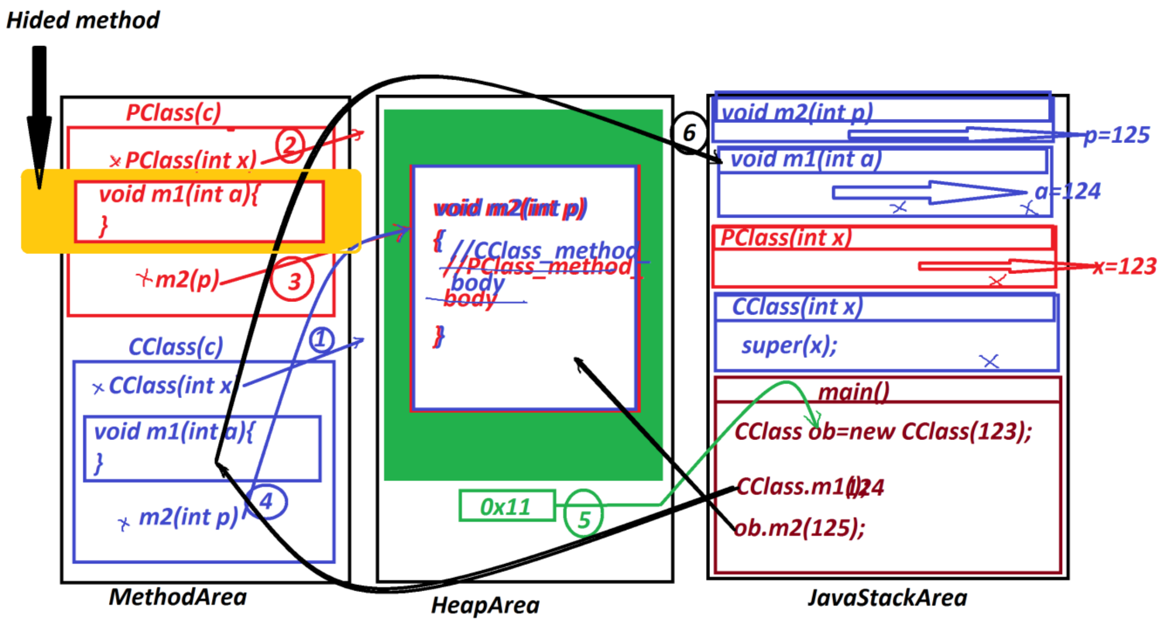
Execution flow of above program:

ClassFiles:

PClass.class

CClass.class

DemoInheritance4.class(MainClass)



==

faq:

define Method Hiding process?

=>The execution control cannot reach PClass Static method when same static method available in CClass,in this process PClass Static method is hided by CClass static method,is known as Method Hiding process.

===

Summary:

(i)Same Instance method signatures in PClass and CClass,is known as Method

Overriding process.

(ii)Same Static method signatures in PClass and CClass,is known as Method Hiding process.

==