Dt : 31/12/2020

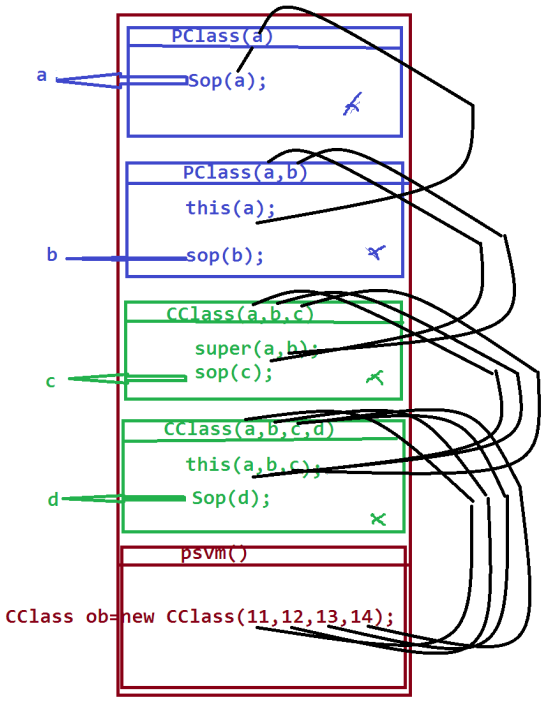
Execution flow of above program:

ClassFiles:

PClass.class

CClass.class

Inheritance9.class



Dt : 1/1/2021

faq:

define Constructor Chaining process:

=>The process of calling the Constructor from the Constructor is known as Constructor Chaining process.

Exp

above program

faq:

define Constructor OverLoading process?

=>More than one Constructor differentiated by their Para_list or Para_type is known as Constructor OverLoading process.

Exp:

above Program

faq:

define Constructor Overriding process?

=>There is no concept of Constructor Overriding process because we cannot have same constructors in PClass and CClass.

Inheritance Case-3:

Static members from the PClass or SuperClass.

Exp program:

```
class PClass
{
        static int a;
        static void m1()
        {
```

```
System.out.println("===PClass m1()===");
System.out.println("The value a:"+a);
        }
        static
        {
System.out.println("===PClass Static block===");
        }
}
class CClass extends PClass
{
        static int b;
        static void m2()
        {
System.out.println("===CClass m2()===");
System.out.println("The value b:"+b);
        }
        static
        {
System.out.println("===CClass Static block===");
        }
}
class Inheritance10 //MainClass
{
        public static void main(String[] args)
        {
```

```
CClass.a=12;
CClass.b=13;
CClass.m1();
CClass.m2();
        }
}
```

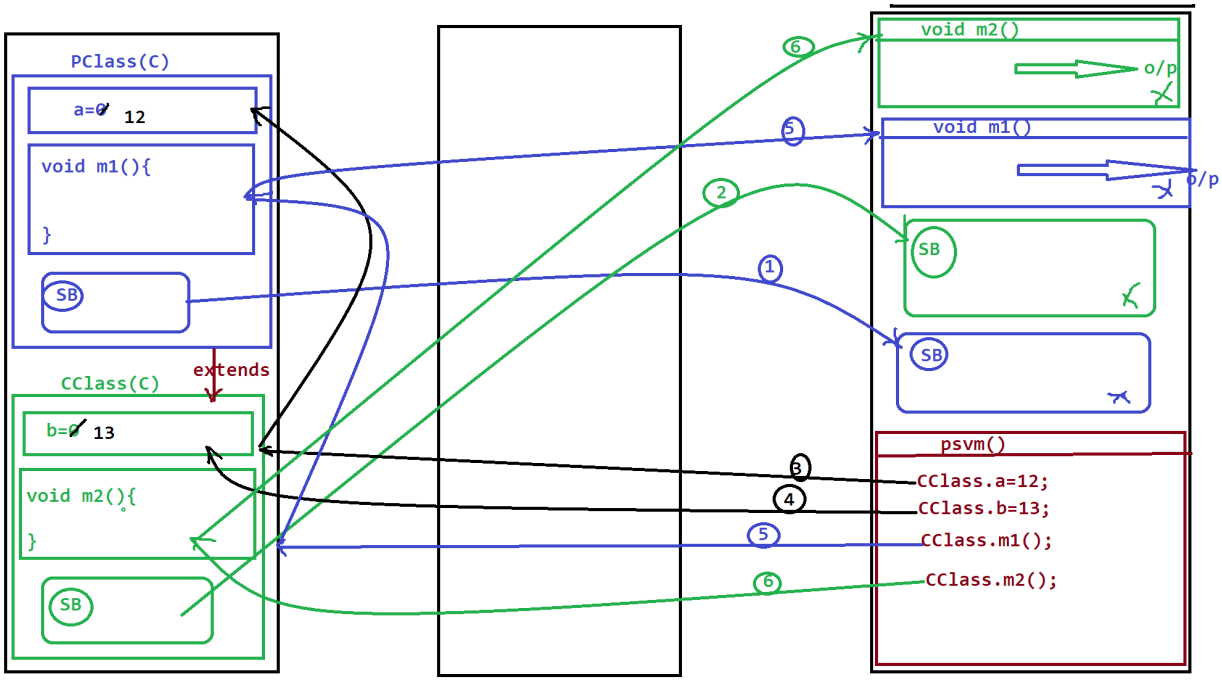
Execution flow of above program:

ClassFiles:

PClass.class

CClass.class

Inheritance10.class



Note:

=>In inheritance process the static members of PClass are available to CClass and can be accessed with the CClass_name.

----

faq:

In wt situation we create object for the class?

=>when we want to access NonStatic members from the class then we create object for the class.