Dt : 30/9/2022

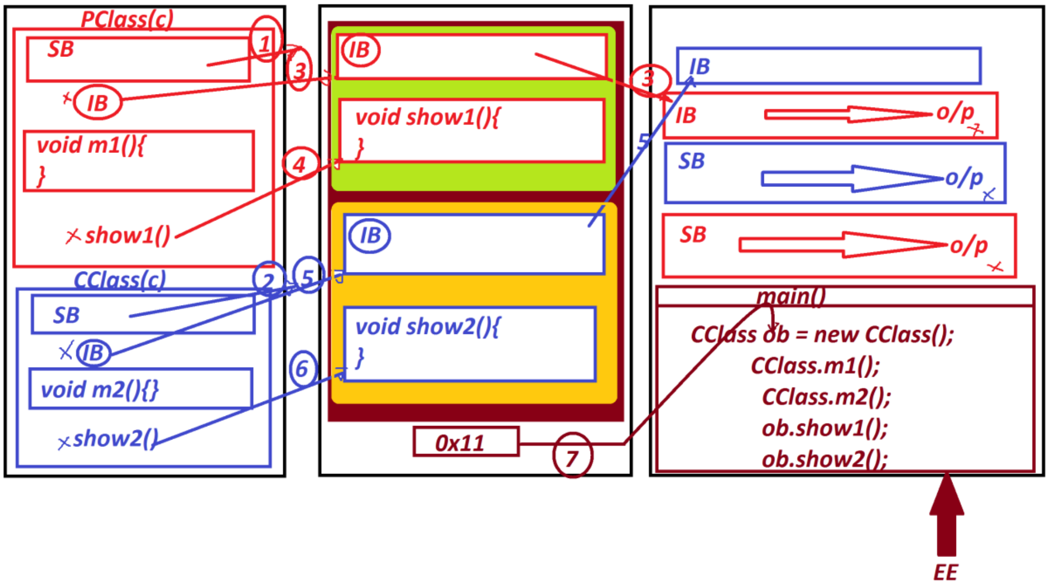
Execution flow of above program:

ClassFiles:

PClass.class

CClass.class

DemoInheritance1.class(MainClass)



Note:

(i)In Normal Inheritance process,object is created for CClass and the object will hold the instance members of PClass.

(ii)In Inheritance process,PClass is loaded to MethodArea first and then the CClass is loaded.

(iii)In Inheritance process while object creation,PClass instance members will get the memory within the object first and then CClass instance

Instance members will get the memory.

Case-2 : Constructors from the PClass/SuperClass

(i)O-parameter Constructor in PClass/SuperClass

=>when we have 0-parameter constructor in PClass then the compiler at compilation stage will add "super()" to the CClass constructor and which is PClass Con_call.

Note:

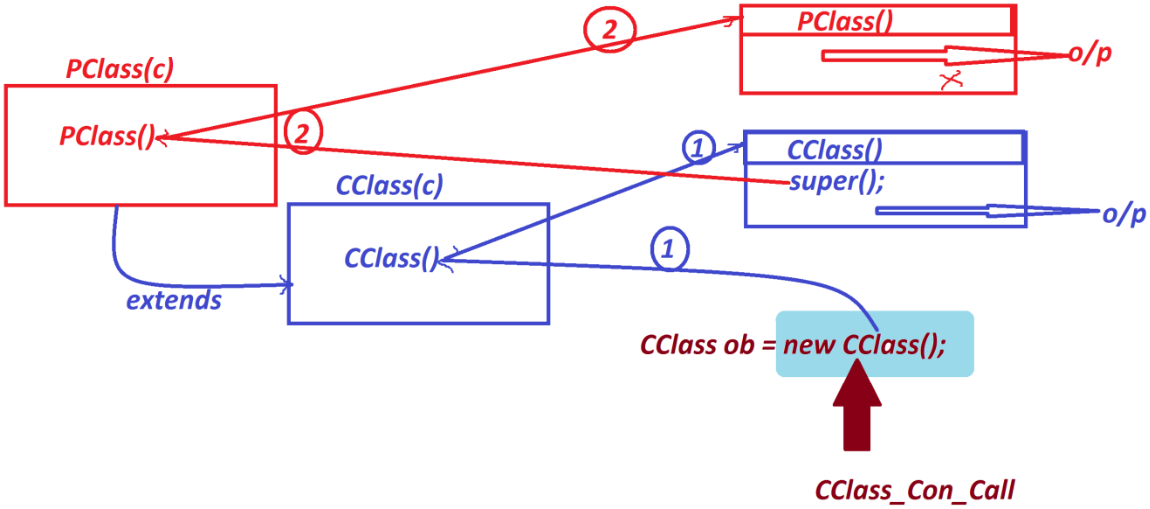
=>In inheritance process,CClass constructor will call PClass Constructor using "super()".

faq:

define Constructor Chaining process?

=>The process of calling one constructor from another constructor is known as Constructor Chaining process or Constructor Interlinking process.

Diagram:



Ex:

PClass.java

```
package test;
public class PClass
{
    public PClass()
    {
        System.out.println("====PClass con====");
    }
}
```

CClass.java

```
package test;

public class CClass extends PClass
{
     public CClass()
     {
      System.out.println("====CClass con===");
     }
}
```

DemoInheritance2.java(MainClass)

```
package maccess;
import test.*;
public class DemoInheritance2 {
     public static void main(String[] args) {
        CClass ob = new CClass();//Con_Call
      }
}
```

o/p:

====PClass con====

====CClass con===

(ii)Parameterized Constructor in PClass/SuperClass

=>when there is Parameterized constructor in PClass/SuperClass,then the programmer must add super() to the ChildClass constructor to pass parameters to the PClass constructor.

Ex:

PClass.java

```
package test;
public class PClass
{
    public PClass(int x)
    {
        System.out.println("====PClass con====");
        System.out.println("The value x:"+x);
    }
}
```

CClass.java

```
package test;
public class CClass extends PClass
{
     public CClass(int p)
```

```
	{
	 super(234);//PClass_Con_Call
	}
}
```

DemoInheritance3.java(MainClass)

```
package maccess;
import test.*;
public class DemoInheritance3 {
	public static void main(String[] args) {
	   CClass ob = new CClass(123);//Con_Call
	}
}
```

o/p:

```
====PClass con====

The value x:123
```

==

**imp*

Define Method Overriding process?

=>The method with same method signature in PClass and CClass,then PClass method is replaced by CClass method while object creation process is known as Method Overriding process.

=>Same method Signature means,

same return_type

same method_name

same para_list

same para_type