***Dt : 26/9/2022***

***\*imp***

***2.Inheritance in Java:***

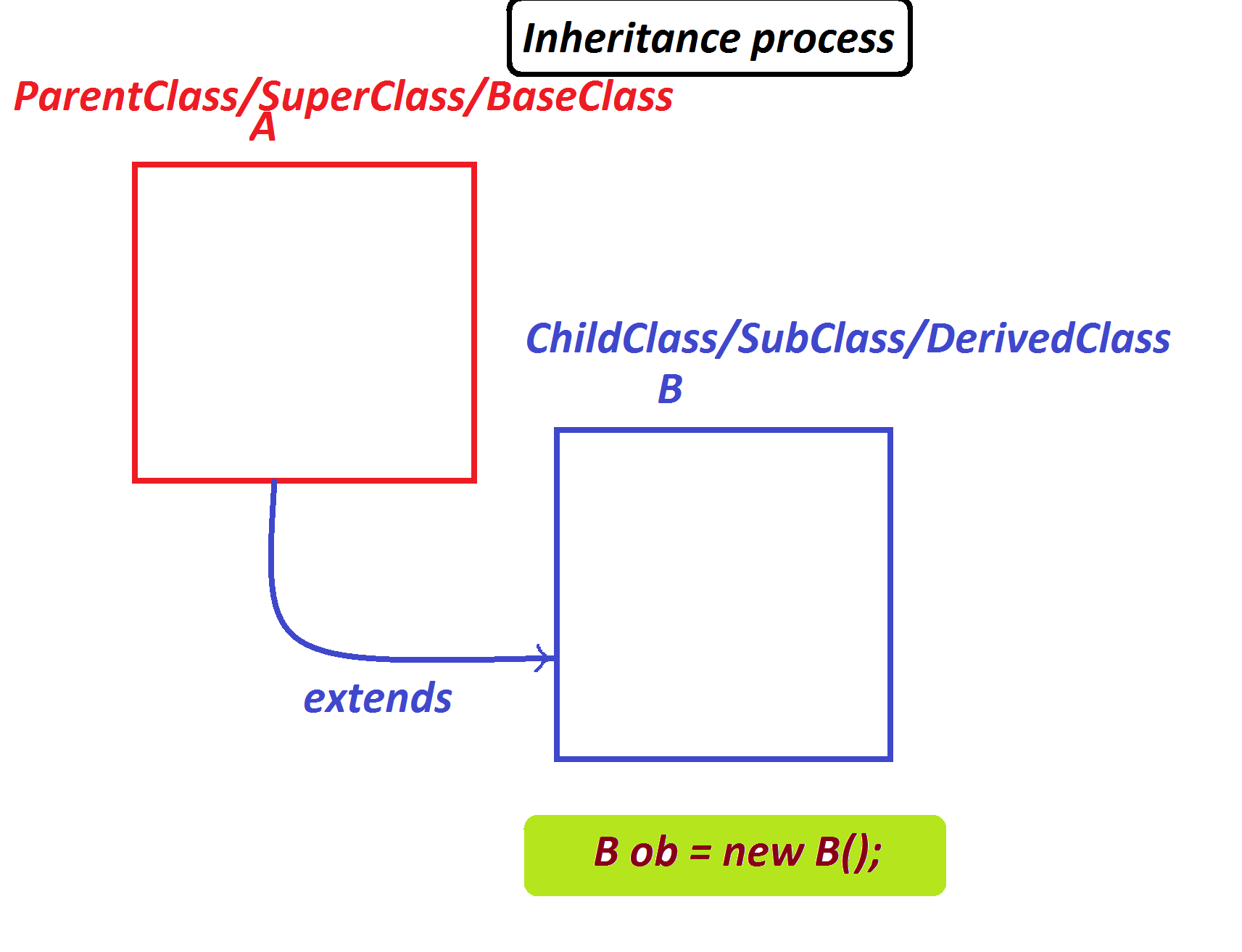
***=>The process of linking two classes with extends keyword is known***

***as Inheritance process.***

***=>In this Inheritance process the features of ParentClass are***

***available to ChildClass.***

***Diagram:***

******

***syntax:***

***class A***

***{***

***//PClass\_body***

***}***

***class B extends A***

***{***

***//CClass\_body***

***}***

***=>In Inheritance process we always create object of ChildClass.***

***B ob = new B();***

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

***Case-1 : Blocks and methods from the ParentClass/SuperClass***

***=>All the static methods of PClass are available to CClass and***

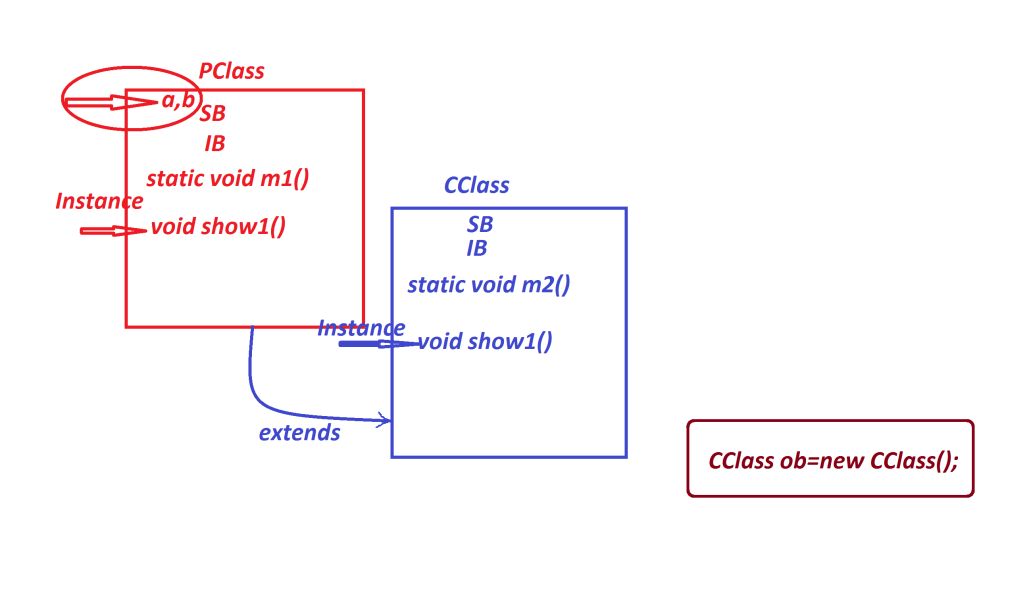
***can be accessed with CClass\_name.***

***=>All the instance methods of PClass will get the memory within***

***the Object of ChildClass and canbe accessed with CClass\_object\_name.***

***=>Static blocks will have highest priority in execution than***

***Instance blocks.***

******

***Ex :***

***PClass.java***

***package test;***

***public class PClass {***

***public static int a=10;***

***public int b=20;***

***static {***

***System.out.println("=====PClass Static block====");***

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

***}***

***{***

***System.out.println("=====PClass Instance block====");***

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

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

***}***

***public static void m1() {***

***System.out.println("====PClass Static m1()=====");***

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

***}***

***public void show1() {***

***System.out.println("====PClass Instance show1()=====");***

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

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

***}***

***}***

***CClass.java***

***package test;***

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

***static {***

***System.out.println("=====CClass Static block====");***

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

***}***

***{***

***System.out.println("=====CClass Instance block====");***

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

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

***}***

***public static void m2() {***

***System.out.println("====CClass Static m2()=====");***

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

***}***

***public void show2() {***

***System.out.println("====CClass Instance shows()=====");***

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

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

***}***

***}***

***DemoInheritance1.java(MainClass)***

***package maccess;***

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

***public class DemoInheritance1 {***

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

***CClass ob = new CClass();//Normal inheritance process***

***CClass.m1();***

***CClass.m2();***

***ob.show1();***

***ob.show2();***

***}***

***}***

***o/p:***

***=====PClass Static block====***

***a:10***

***=====CClass Static block====***

***a:10***

***=====PClass Instance block====***

***a:10***

***b:20***

***=====CClass Instance block====***

***a:10***

***b:20***

***====PClass Static m1()=====***

***a:10***

***====CClass Static m2()=====***

***a:10***

***====PClass Instance show1()=====***

***a:10***

***b:20***

***====CClass Instance shows()=====***

***a:10***

***b:20***

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

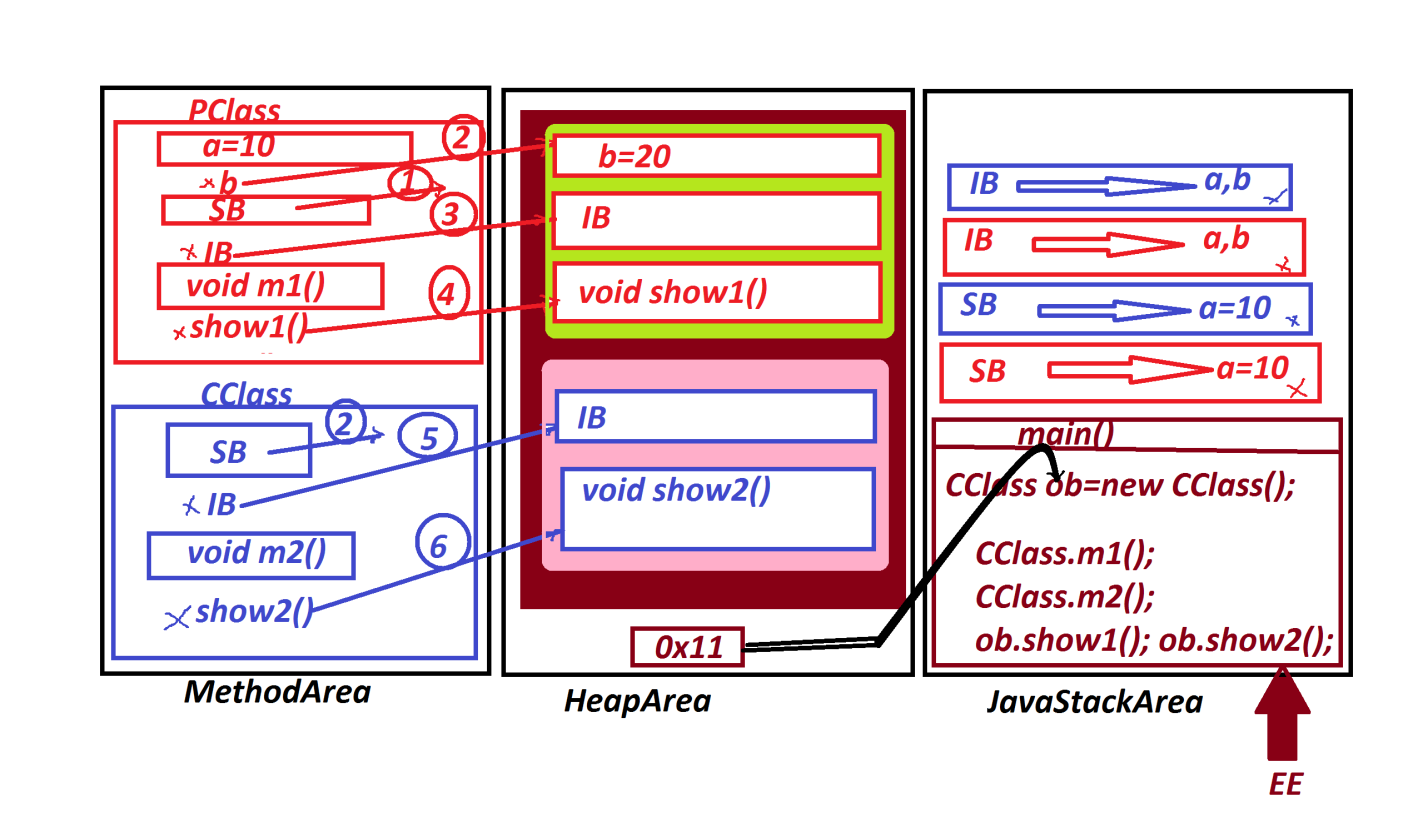
***Execution flow of above program:***

***ClassFiles:***

***PClass.class***

***CClass.class***

***DemoInheritance1.class***

******

***Note:***

***(i)In Inheritance process PClass is loaded onto method area first***

***and then the CClass is loaded.***

***(ii)In Inheritance process one object is created for CClass and***

***the object will hold all the Instance members of PClass and CClass.***

***(iii)In Inheritance process while object creation PClass Instance***

***members will get the memory first and then CClass Instance members***

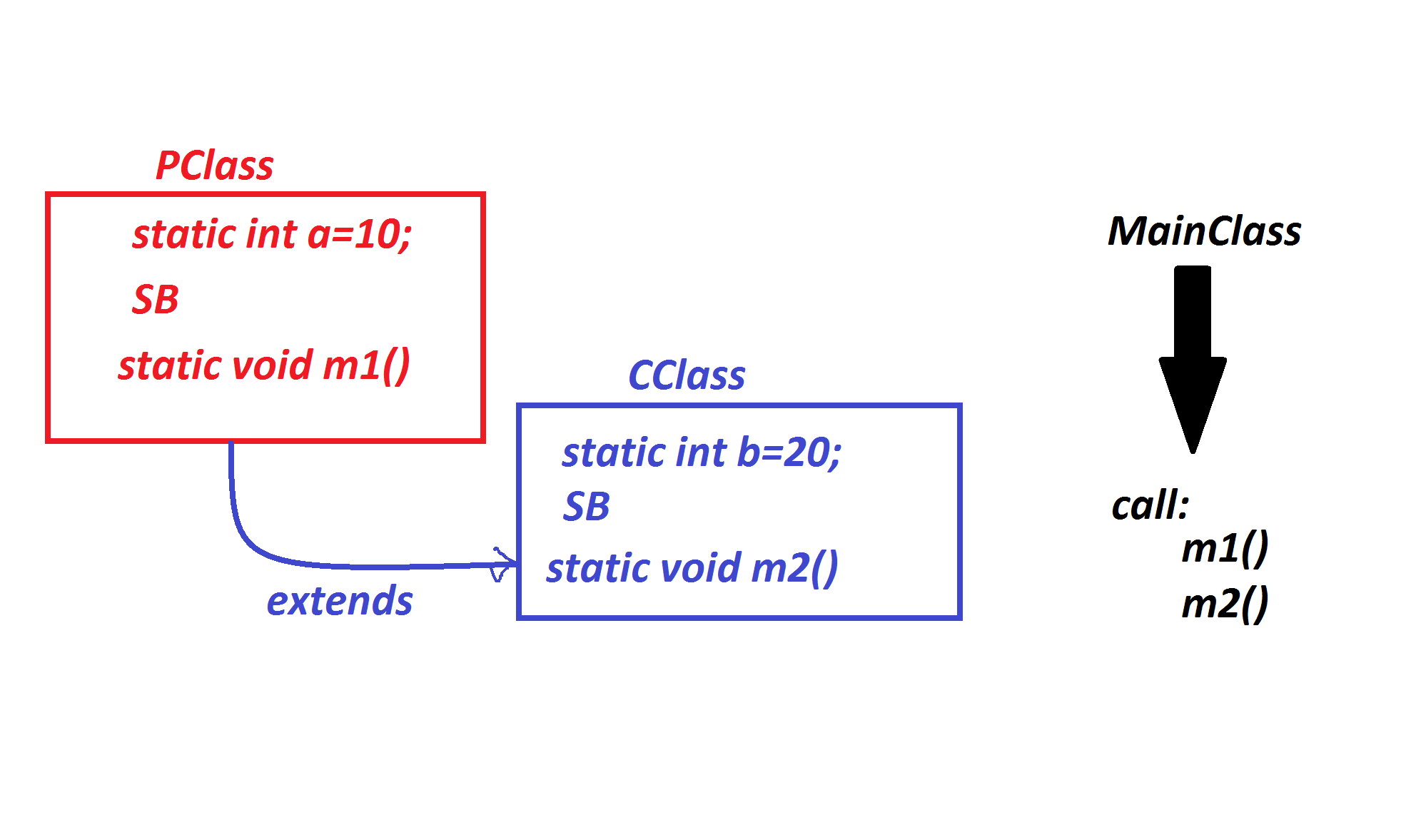
***will get the memory.***

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

***Assignment:***

***Construct the program for the following layout and analyze the***

***output:***

******

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