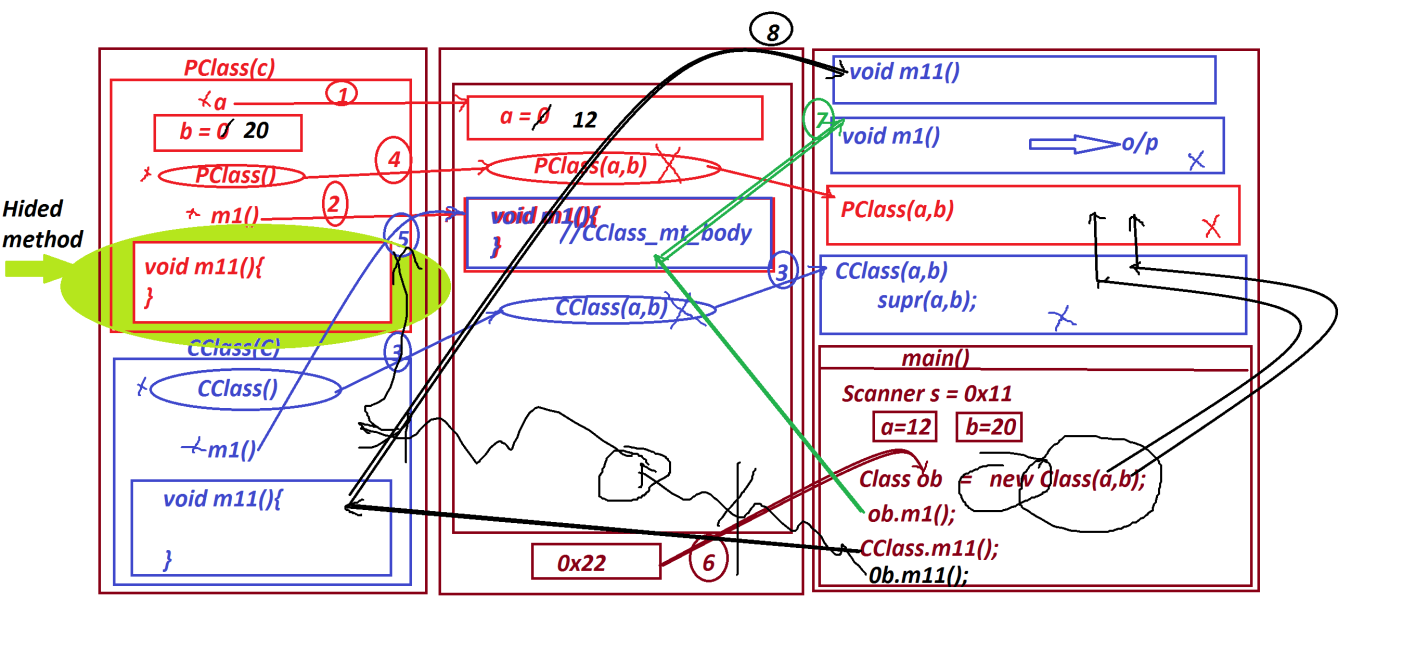
***Dt : 21/9/2022***

***Diagram:***

******

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

***Note:***

***=>Instance method Overriding process is posible.(available)***

***=>There is no concept of Static method Overriding process,because***

***the methods are binded and available in classes.***

***=>There is no Concept of Constructor Overriding process,because***

***we cannot have same constructor names in PClass and CClass.***

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

***faq:***

***define Method Hiding process?***

***=>when we have same static method signature in PClass and CClass,***

***then the execution control cannot reach PClass method for execution,***

***is known as Method Hiding process,in which PClass method is hided***

***by CClass method.***

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

***Summary:***

***=>Same instance method signature in PClass and CClass is known as***

***Method Overriding process.***

***=>Same static method signature in PClass and CClass is known as***

***Method Hiding Process.***

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

***faq:***

***can we access static methods using Object reference?***

***=>Yes,we can access static methods using Object reference,because***

***the Object reference is generated from the class and belongs to***

***class.***

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

***faq:***

***define Generalization process?***

***=>The process in which one object is created holding all the***

***members of PClass and only Overriding members from CClass is known***

***as Generalization process.***

***=>we use the following syntax to perform Generalization process:***

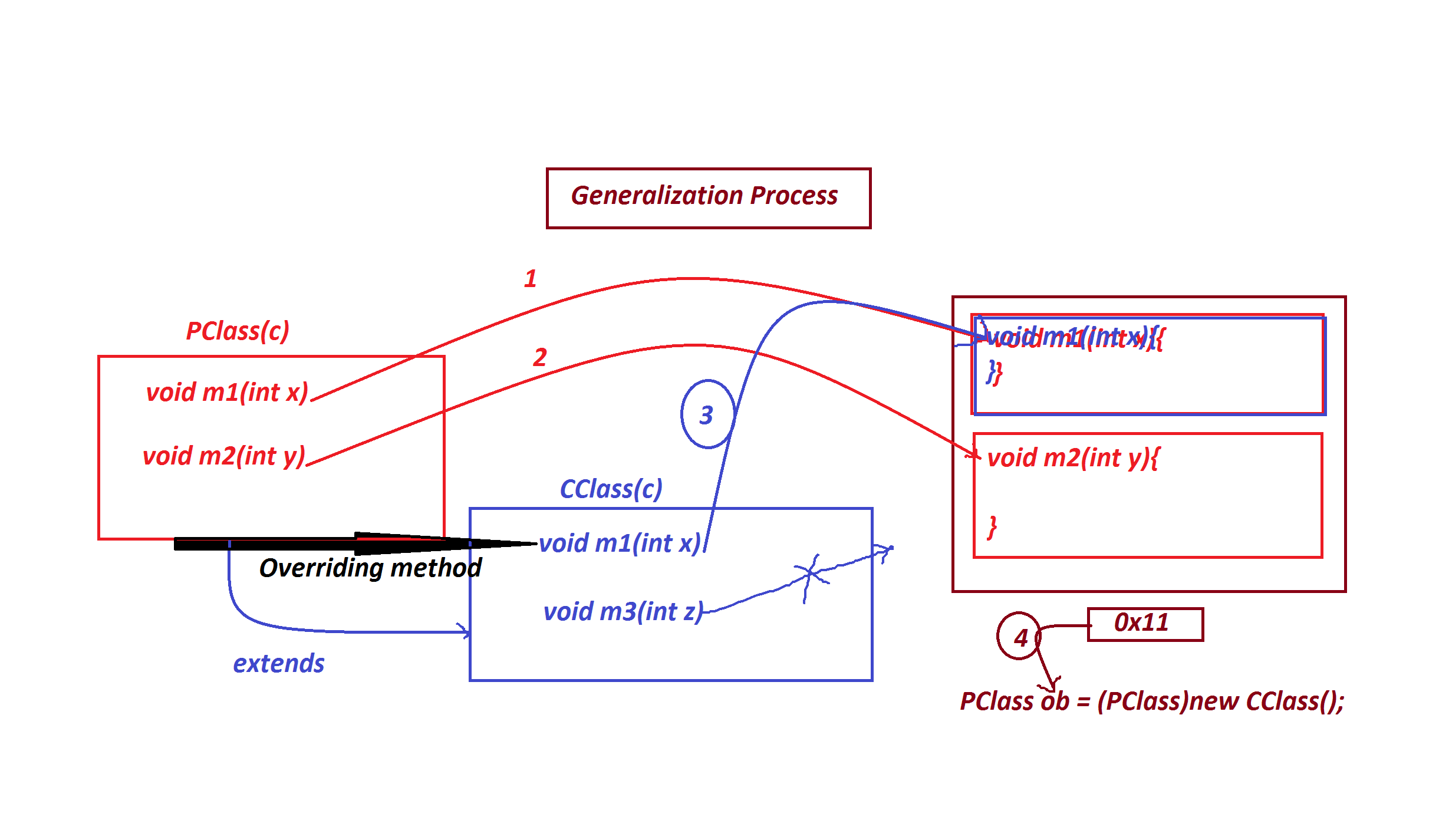
***syntax:***

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

***=>This generalization process is also known as UpCasting process***

***or Widening process.***

***Diagram:***

******

***PClass.java***

***package test;***

***public class PClass {***

***public void m1(int x) //Overrided\_method***

***{***

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

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

***}***

***public void m2(int y) {***

***System.out.println("====PClass m2(y)===");***

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

***}***

***}***

***CClass.java***

***package test;***

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

***public void m1(int x)//Overriding method***

***{***

***System.out.println("====CClass m1(x)===");***

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

***}***

***public void m3(int z)//NonOverriding method***

***{***

***System.out.println("====CClass m3(z)===");***

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

***}***

***}***

***DemoInheritance4.java(MainClass)***

***package maccess;***

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

***public class DemoInheritance4 {***

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

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

***ob.m1(12);***

***ob.m2(23);***

***// ob.m3(23); //Error***

***}***

***}***

***o/p:***

***====CClass m1(x)===***

***The value x:12***

***====PClass m2(y)===***

***The value y:23***