***Dt : 30/9/2022***

***faq:***

***Generalization process:***

***=>In Generalization process,one object created holding all the***

***members of PClass and only Overriding members from the CClass.***

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

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

***Note:***

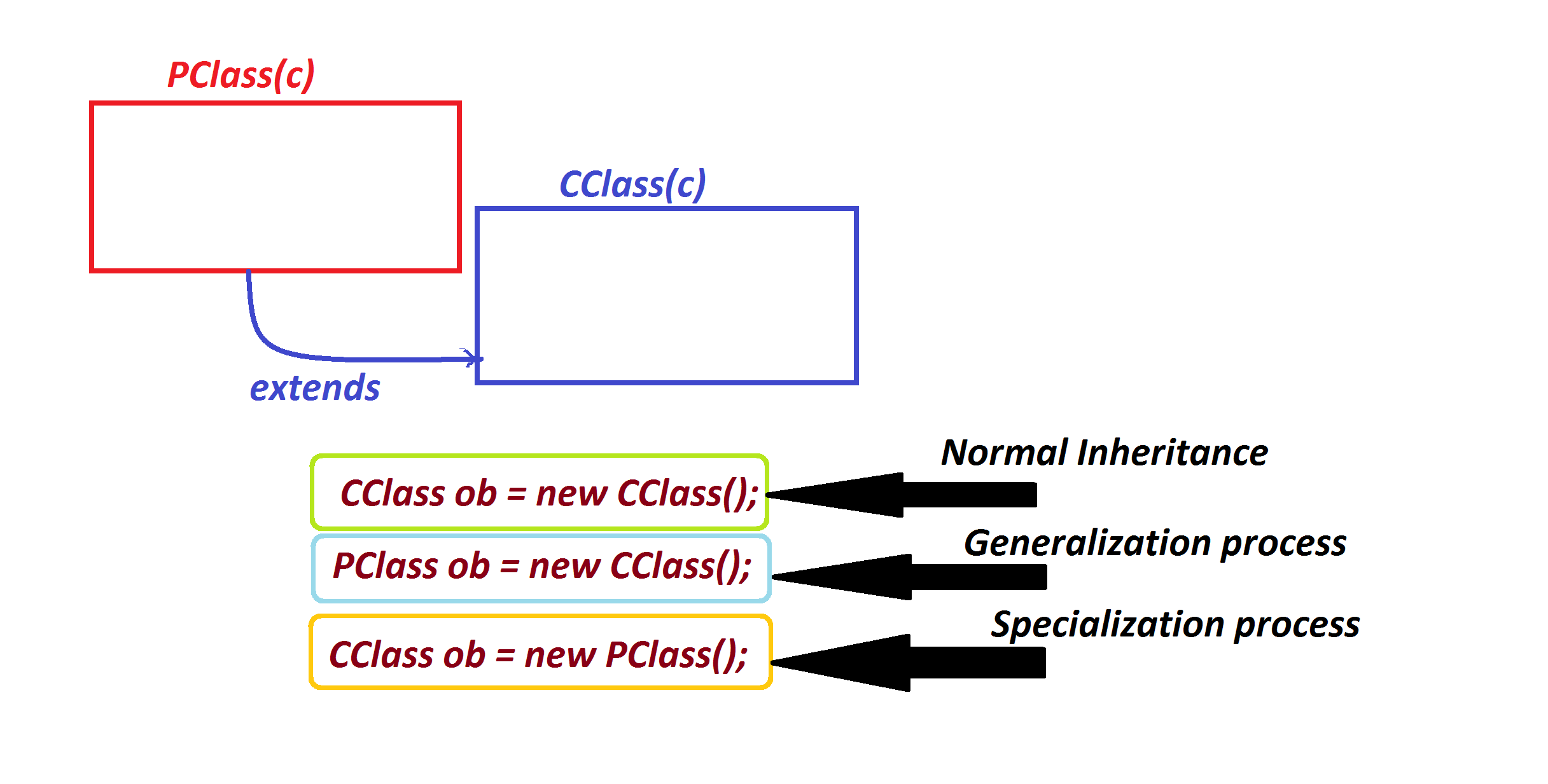
***=>In this Generalization process,object created for CClass but***

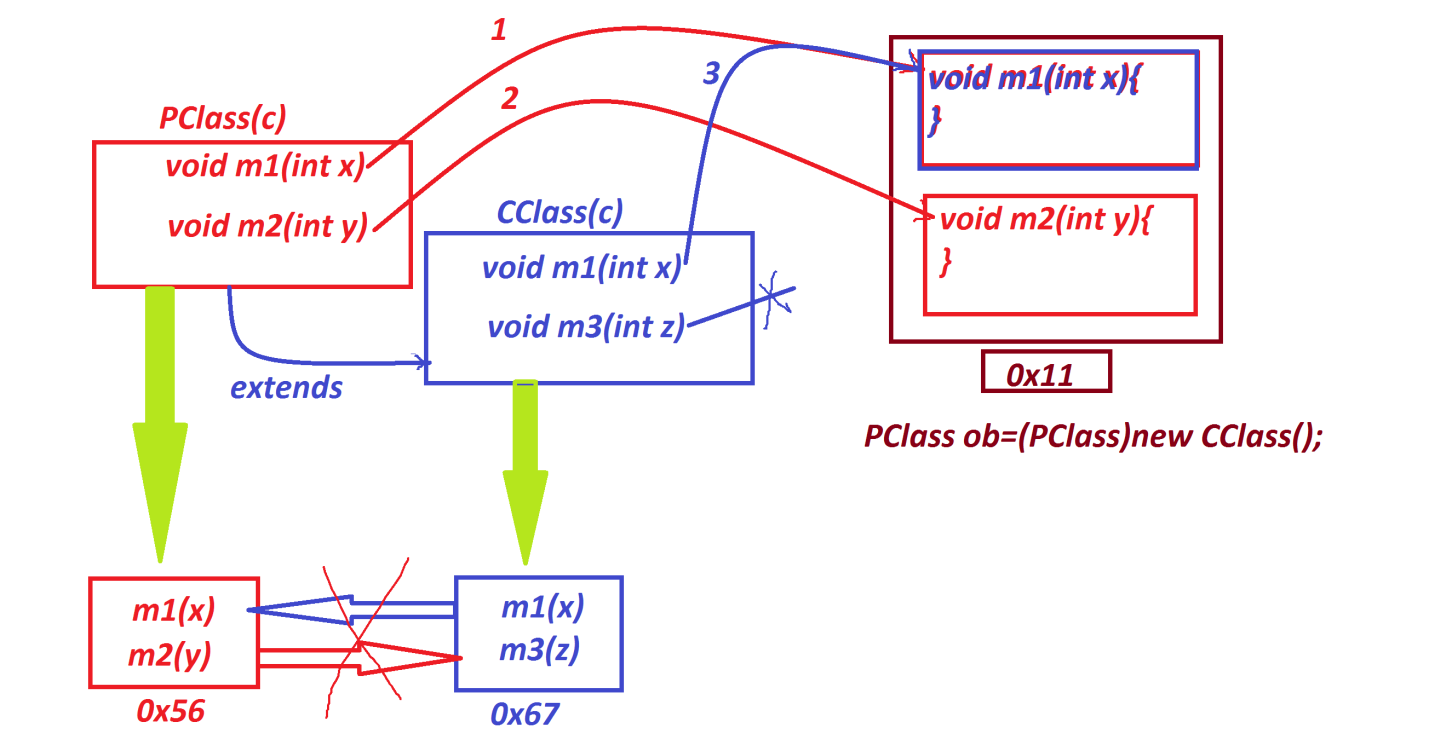
***Converted into PClass.***

***=>Generalization process is also known as UpCasting process or***

***Widening process or Implicit TypeCasting process.***

***Diagram:***

******

******

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

***faq:***

***Specialization process:***

***=>The process in which CClass is constructed based on one feature***

***of PClass is known as Specialization process.***

***=>we use the following syntax to construct Specialization process:***

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

***Note:***

***=>This Specialization process is also known as DownCasting process***

***or Narrowing process or Explicit TypeCasting process.***

***=>In Specialization process,the PClass must be Pre-defined class***

***from JavaLib,if not raises "java.lang.ClassCastException".***

***Ex:***

***Cloning process***

***Serialization-DeSerialization process***

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

***Ex:***

***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{***

***@Override***

***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)//Non\_Overriding method***

***{***

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

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

***}***

***}***

***DemoInheritance10.java(MainClass)***

***package maccess;***

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

***public class DemoInheritance10 {***

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

***System.out.println("\*\*\*\*Generalization process\*\*\*\*");***

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

***ob.m1(11);***

***ob.m2(12);***

***//ob.m3(13);//Error***

***System.out.println("\*\*\*\*Specialization process\*\*\*\*");***

***CClass ob2 = (CClass)new PClass();***

***ob2.m1(11);***

***ob2.m2(12);***

***ob2.m3(13);//Error***

***}***

***}***

***o/p:***

***\*\*\*\*Generalization process\*\*\*\****

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

***The value x:11***

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

***The value y:12***

***\*\*\*\*Specialization process\*\*\*\****

***Exception in thread "main" java.lang.ClassCastException:***

***class test.PClass cannot be cast to class test.CClass***

***(test.PClass and test.CClass are in unnamed module of loader 'app')***

***at maccess.DemoInheritance10.main(DemoInheritance10.java:11)***

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

***\*imp***

***Types of Inheritances:***

***=>Inheritances are categorized into the following:***

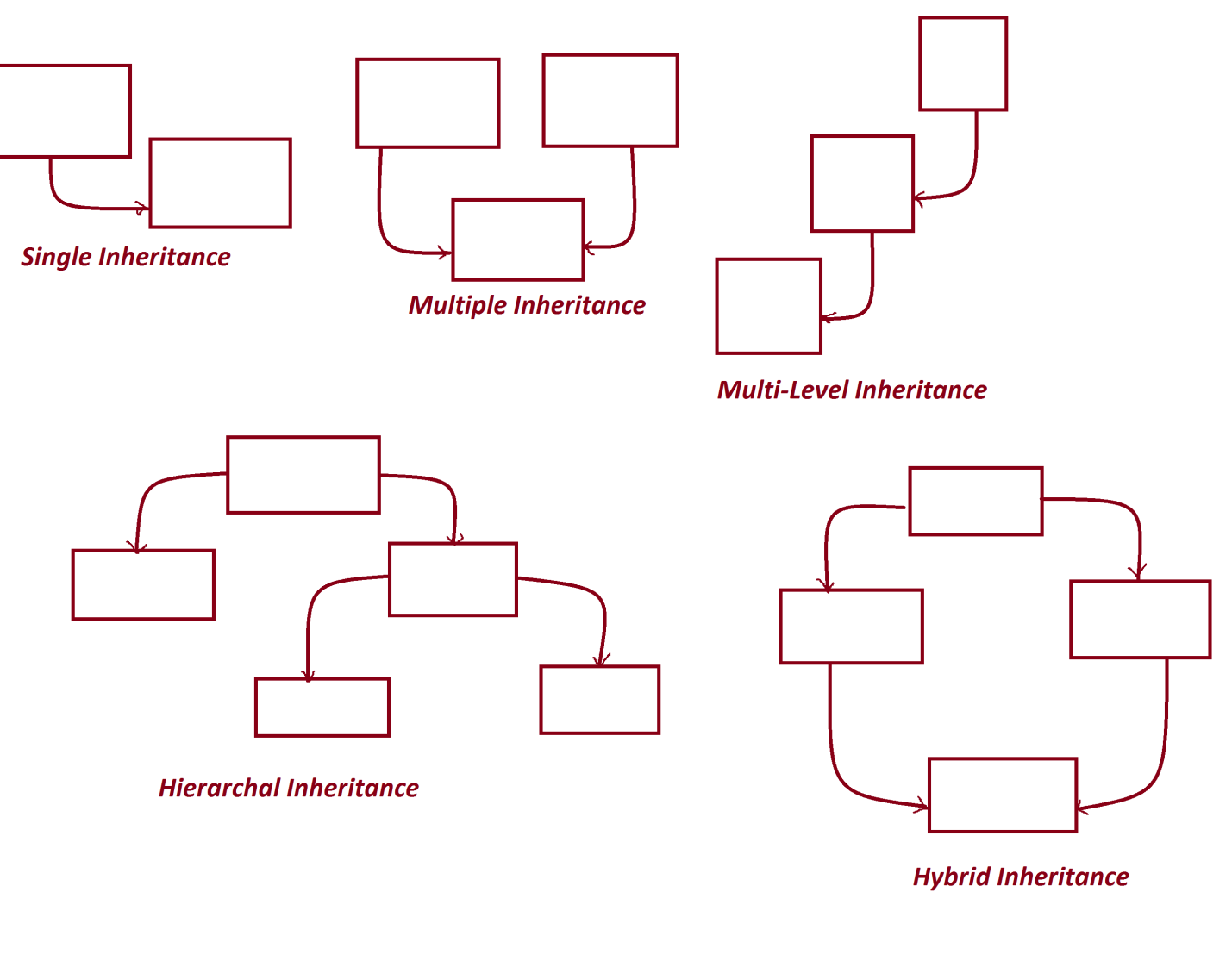
***1.Single Inheritance***

***2.Multiple Inheritance***

***3.Multi-Level Inheritance***

***4.Hierarchal Inheritance***

***5.Hybrid Inheritance***

******