***Dt : 19/9/2022***

***Note:***

***=>Reading Numeric data after String data from console will not***

***make problem in reading data,but Reading String data after numeric***

***data will skip the string data reading,because the buffer will hold***

***enter-key-information and skips the data.***

***=>This problem can be overcomed using parse methods available from***

***WrapperClasses:***

***byte b = Byte.parseByte(s.nextLine());***

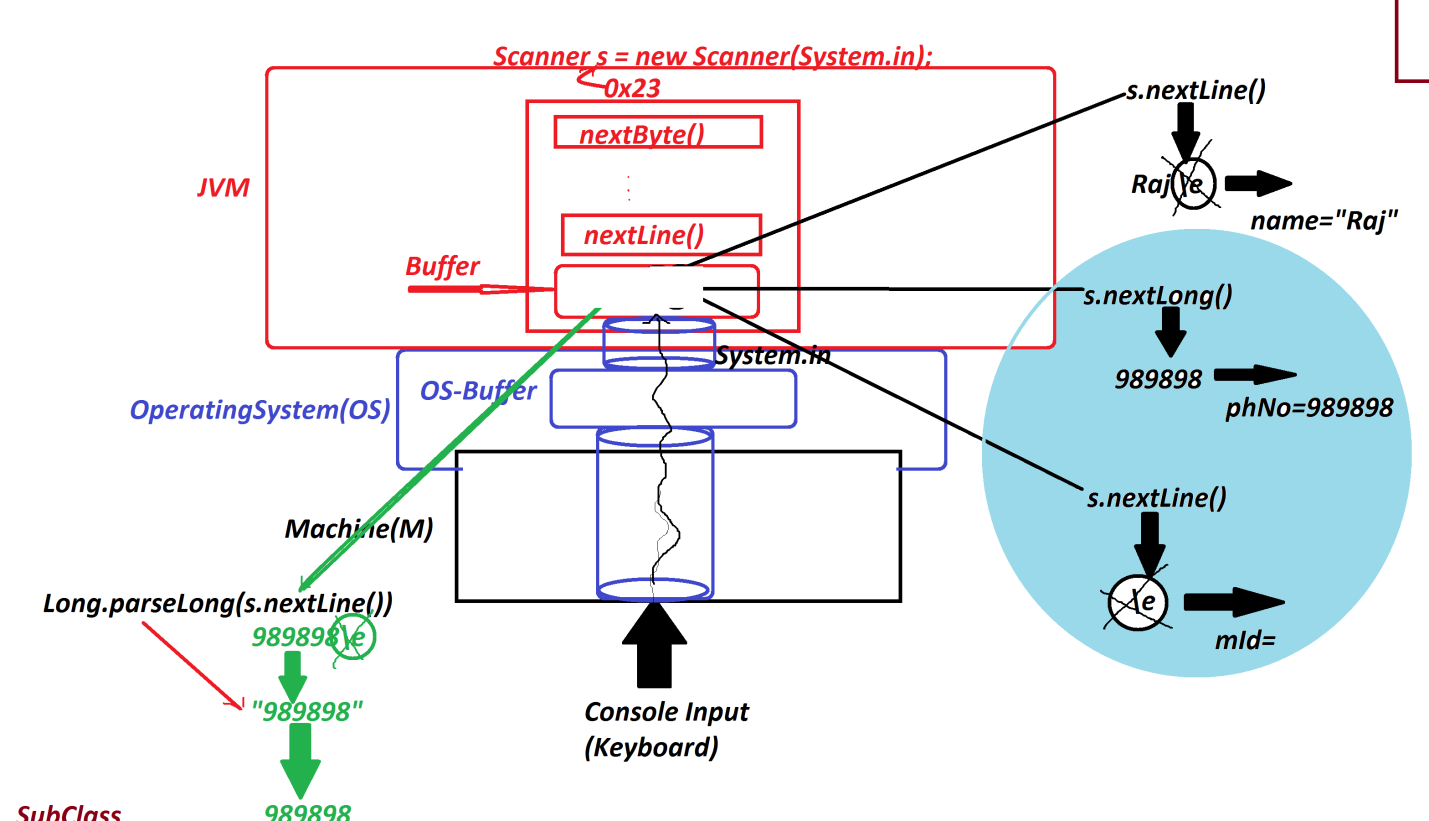
***short sh = Short.parseShort(s.nextLine());***

***int i = Integer.parseInt(s.nextLine());***

***long l = Long.parseLong(s.nextLine());***

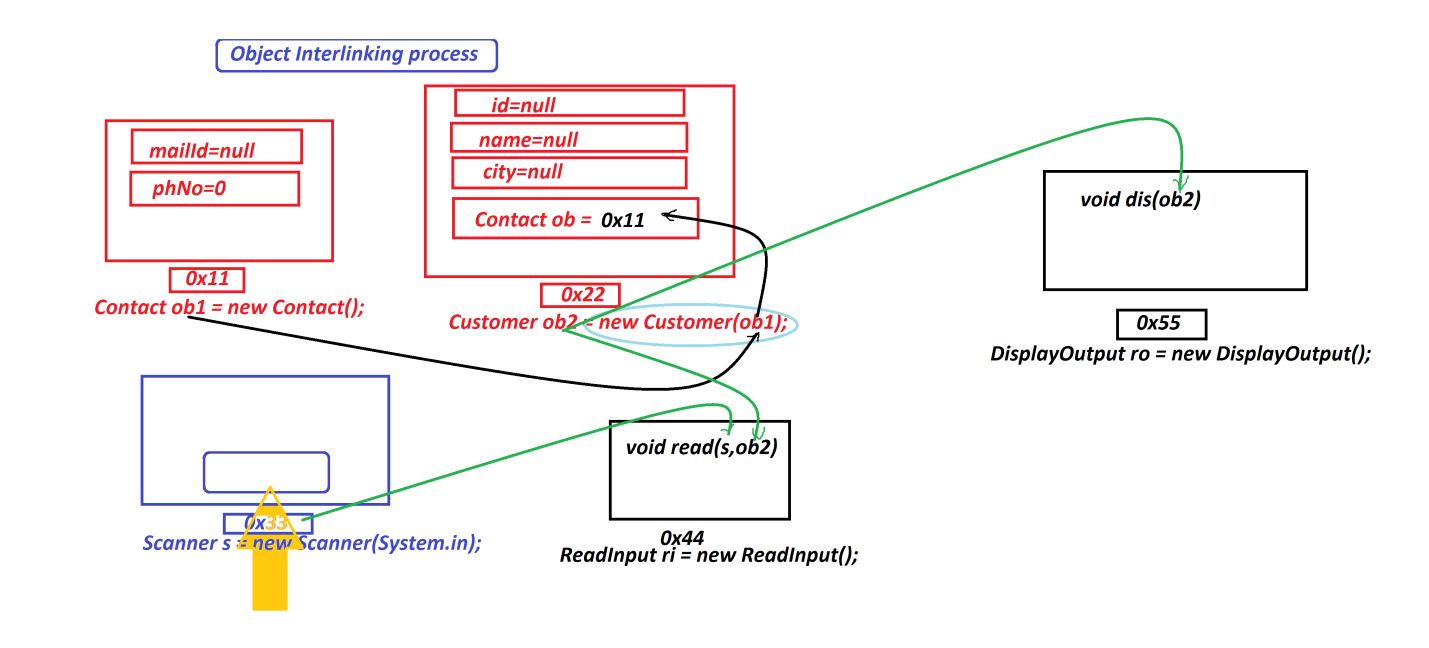
***float f = Float.parseFloat(s.nextLine());***

***double d = Double.parseDouble(s.nextLine());***

******

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

***Diagram of References:***

******

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

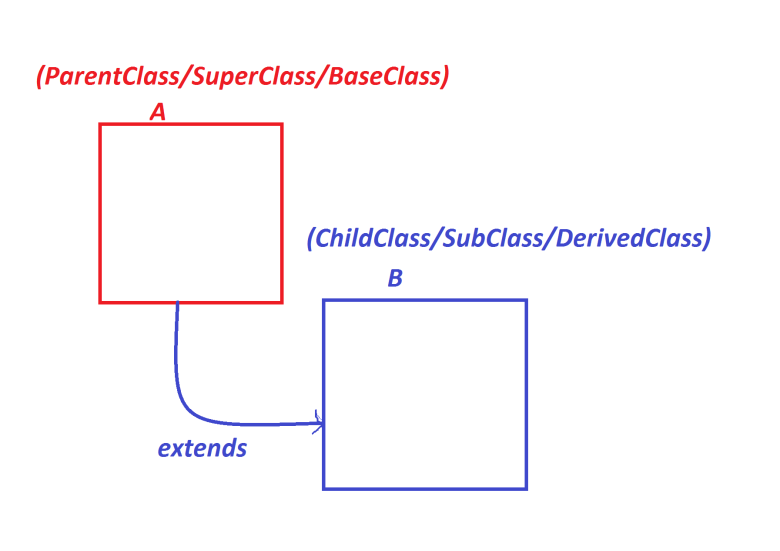
***\*imp***

***2.Inheritance:***

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

***known as Inheritance process.***

***Diagram:***

******

***syntax:***

***class A***

***{***

***//body***

***}***

***class B extends A***

***{***

***//body***

***}***

***=>In Inheritance process the members of ParentClass(A) are available***

***to ChildClass(B),in this process we create object for ChildClass.***

***syntax:***

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

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

***Case-1 : Methods and Blocks from the ParentClass or SuperClass:***

***Ex :***

***PClass.java***

***package test;***

***public class PClass {***

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

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

***static***

***{***

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

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

***}***

***{***

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

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

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

***}***

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

***{***

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

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

***}***

***public void m11()***

***{***

***System.out.println("====PClass Instance m11()====");***

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

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

***}***

***}***

***CClass.java***

***package test;***

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

***public int x=100;***

***public static int y=200;***

***static***

***{***

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

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

***}***

***{***

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

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

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

***}***

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

***{***

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

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

***}***

***public void m22()***

***{***

***System.out.println("====CClass Instance m22()====");***

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

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

***}***

***}***

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

***package maccess;***

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

***public class DemoInheritance1 {***

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

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

***CClass.m1();//PClass\_Static\_method***

***ob.m11();//PClass\_Instance\_method***

***CClass.m2();//CClass\_static\_method***

***ob.m22();//CClass\_Instance\_method***

***}***

***}***

***o/p:***

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

***The value b:20***

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

***The value y:200***

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

***The value a:10***

***The value b:20***

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

***The value x:100***

***The value y:200***

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

***The value b:20***

***====PClass Instance m11()====***

***The value a:10***

***The value b:20***

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

***The value y:200***

***====CClass Instance m22()====***

***The value x:100***

***The value y:200***

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