**If the method in base is private and method in the subclass is public. Can I override the method? Justify.**

class ElectronicDevices

{

private void On()

{

System.out.println("ON");

}

}

class Tv extends ElectronicDevices

{

public void On()

{

super.On();

System.out.println("The device is being turned On");

}

}

class TestInheritance

{

public static void main(String[] args)

{

Tv obj=new Tv();

obj.On();

}

}

Output

F:\CTS\Java>javac TestInheritance.java

TestInheritance.java:12: error: On() has private access in ElectronicDevices

super.On();

^

1 error

Explanation: Since we declared the base class method as private, the subclass will not be able to override the method. So the method will not get over rided. This is called **METHOD HIDING**.

**If we are creating objects for the subclasses, WAP to count the total no of objects being created**

**package** com.cts.findobjects;

**public** **class** Employee {

**static** **int** *counter*;

**public** Employee()

{

*counter*++;

}

**public** **void** Count()

{

System.***out***.println("The no of objects="+*counter*);

}

}

**package** com.cts.findobjects;

**public** **class** Manager **extends** Employee {

}

**package** com.cts.findobjects;

**public** **class** Admin **extends** Employee {

}

**package** com.cts.findobjects;

**public** **class** Test {

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

// **TODO** Auto-generated method stub

Manager ob1=**new** Manager();

Manager ob2=**new** Manager();

Admin ob3=**new** Admin();

Admin ob4=**new** Admin();

ob3.Count();

}

}

Output

The no of objects=4

**If my subclasses are TV and DVD player, what will be the base class? Implement inheritance and overriding for this example.**

**public** **class** ElectronicDevices {

**public** **void** On() {

System.***out***.println("ON");

}

**public** **void** Off() {

System.***out***.println("Off");

}

}

**public** **class** Tv **extends** ElectronicDevices{

**public** **void** ChangeChannels() {

System.***out***.println("Changing channels");

}

}

**public** **class** Dvd {

**public** **void** On() {

System.***out***.println("the DVD is on now");

}

}

}

**public** **class** Test {

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

Tv ob1=**new** Tv();

Dvd ob2=**new** Dvd();

ob1.Off();

ob2.On();

}

}

Output

Off

the DVD is on now

**Implement inheritance and overriding for Student(base class), day scholar and residential.**

**public** **class** Student {

**public** **void** Write() {

System.***out***.println("Writing");

}

**public** **void** Read() {

System.***out***.println("Reading");

}

}

**public** **class** DayScholar **extends** Student {

**public** **void** walk() {

System.***out***.println("Walking");

}

}

**public** **class** Residential **extends** Student {

**public** **void** Read() {

System.***out***.println("Reading from Hostel");

}

}

**public** **class** Test {

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

// **TODO** Auto-generated method stub

DayScholar ob1=**new** DayScholar();

Residential ob2=**new** Residential();

ob1.Write();

ob2.Read();

}

}

Output

Writing

Reading from Hostel

**How will the overriding of equals method in object class will help?**

<http://users.csc.calpoly.edu/~gfisher/classes/102/info/howToOverrideEquals.html>

**Can I make constructor as**

**Final**?

NO

public class Finalcon {

int length;

int width;

final Finalcon(int x, int y)

{

this.length=x;

this.width=y;

}

public static void main(String[] args) {

// TODO Auto-generated method stub

Finalcon f=new Finalcon(10,20);

System.out.println(f.length);

}

}

Output

F:\CTS\Java>javac Finalcon.java

Finalcon.java:4: error: modifier final not allowed here

final Finalcon(int x, int y)

^

1 error

**Private?**

For a normal java class, it is possible.

public class Finalcon {

int length;

int width;

private Finalcon(int x, int y)

{

this.length=x;

this.width=y;

}

public static void main(String[] args) {

// TODO Auto-generated method stub

Finalcon f=new Finalcon(10,20);

System.out.println(f.length);

}

}

Output

F:\CTS\Java>javac Finalcon.java

F:\CTS\Java>java Finalcon

10

But in case of inheritance, if the base class will have a constructor with modifier private, it cant be used by the objects of subclasses.