**Questions: (08/01/2020)**

1. If method in base class is private and if method in derived class is public, can we override?

Ans. You cannot override a method from base class which is private because base method signature will not be visible to the derived class.

For example:

public class B {

private void don() {

System.out.println(" private");

}

public static void main(String[] args) {

B t = new A();

t.don(); -> This will print “private” as private method cannot be overridden.

}

}

class A extends B {

@Override

public void don() {

System.out.println("hoho public");

}

}

1. Whenever we try to create 2 objects in Manager and 3 objects in Admin, It should say 5 Employee objects created?

Ans. Refer Github for code

1. If my subclass name are TV and DVD Player, what is base class? Implement inheritance and overriding?

Ans. Refer Github for code

1. Implement Student as Base class, subclass are DayScholar and Residential. Implement inheritance and create apt method?

Ans. Refer Github for code

1. Take Employee class example. Employee class has empId and EmpName. Admin has inventoryQuantity and cadre. Manager has numberofSubordinates. Whenever we create object of Admin, it should print employee variable and admin variable. Similarly for manager.

Ans. Refer Github for code

1. Can we make constructor as private?

Ans. Yes, we can create a private constructor. Private constructors are used to serve singleton class. Singleton class are those classes which are restricted to a single object. Private constructors restricts the creation of single object at a time.

1. Can we make constructor as final?

Ans. No, we can’t make a constructor as final because final is used for method when we don’t want anyone to override it. Since constructors are never overridden, so there is no use to make it final.

1. Java Ques:

a)

class A{

private void method(){

System.out.println("Hii");

}

}

class B extends A{

public void method(){

System.out.println("Hello");

}

}

class Test{

public static void main(String args[]){

new A().method();

new B().method();

}

}

OUTPUT: This will print “Hello”, but it is not method overriding rather it is Method hiding.

b)

class A{

public void method(){

System.out.println("Hii");

}

}

class B extends A{

public final void method(){

System.out.println("Hello");

}

}

class Test{

public static void main(String args[]){

new B().method();

}

}

OUTPUT: will print “hello” because base class is not final and hence can be overridden.