JAVA 8/1/20

1.ReadOnlyVariable

**public** **class** ReadOnly {

**private** **int** id;

**public** ReadOnly() {

}

**public** **int** getid(**int** id) {

**return** id;

}

}

**public** **class** Demo {

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

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

ReadOnly R1= **new** ReadOnly();

System.***out***.println(R1.getid(54));

}

}

2. If Method in base class is private and in sub class it is public ca we override it ?

Sol:- No, because private methods cannot be overridden. In derived class the private method is not visible.

3. base Employee class == id,name sub class Manager== no of subordinates

Admin == inventory

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

**int** id;

String name;

**void** id(**int** i) {

System.***out***.println("id is"+i);

}

**public** Employee() { }

**public** Employee(**int** id, String name) {

**this**.id=id;

**this**.name=name;

}

}

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

**public** **void** NoOfSubOrdinates (**int** i) {

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

System.***out***.println("Total is "+i);

}

}

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

**public** **void** InnQua(**int** t) {

System.***out***.println("Inventory Quantity is " +t);

}

**public** Admin() {}

**public** Admin(**int** id, String name) {

// **TODO** Auto-generated constructor stub

}

}

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

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

Admin e = **new** Admin ();

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

e.id(1111);

e.name("Gaurav");

ob1.NoOfSubOrdinates(5);

e.InnQua(45);

}

}

4. What is the purpose of overriding equals methods ?

Sol:- The equal method of object class check reference hence to make that

Method to check logical equality you must override it.

5. Can we make constructor as private?

Sol:- Yes, we can make a constructor private. When a class need to prevent caller to make objects then private constructor is suitable

6. can we write final before constructor ?

Sol:- constructors cannot be inherited in java and also cannot be overridden no no need to write final before constructor.