Q1. 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.

Q2. 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.

Q3 What is the purpose of overriding equals and hashcode methods of Object class?

Ans. equals and hashCode in Java are two fundamental methods which are declared in Object class and part or core Java library. equals() method is used to [compare Objects](http://javarevisited.blogspot.sg/2011/06/comparator-and-comparable-in-java.html) for equality while hashCode is used to generate an integer code corresponding to that object.  
  
equals method is also used to avoid duplicates on HashSet and other Set implementation and every other place where you need to compare Objects.  
  
Since [HashMap and Hashtable](http://javarevisited.blogspot.sg/2010/10/difference-between-hashmap-and.html) in Java rely on equals() and hashCode() method for comparing keys and values, so java recommends to override hashcode() and equals() method.

And equals method in Java must follow its contract with hashcode method in Java as stated below.

1) If two objects are equal by equals() method then there hashcode must be same.

2) If two objects are not equal by equals() method then there hashcode could be same or different.

* 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.