

PERFECT SINGLETON

Designing perfect singleton class:-

```
1. package com.nit.perfect;
2.
3. public class Printer {
4.
5.     public static Printer instance;
6.     private static boolean flag=false;
7.
8.     //0 param constructor
9.     //strict constructor for reflection api
10.    private Printer() {
11.        if(flag==true) {
12.            throw new IllegalArgumentException("object alreedy created");
13.        }
14.        flag=true;
15.        System.out.println("0 param constructor");
16.    }
17.
18.    //static factory method
19.    public static Printer getInstance() {
20.        if(instance==null) {
21.            synchronized (Printer.class) {    //class level locking
22.                if(instance==null) {            //double null check
23.                    instance=new Printer();
24.                }
25.            }
26.        }
27.        return instance;
28.    }
29.
30.    //override clone method from object
31.    @Override
32.    public Object clone() throws CloneNotSupportedException {
33.        System.out.println("Printer.clone()");
34.        return super.clone();
35.    }
36.
37.    //by using deserialization
38.    public Object readResolve() {
39.        System.out.println("Printer.readResolve()");
40.        throw new IllegalArgumentException("object alreedy created");
41.    }
42. }
```

PERFECT SINGLETON

VIKAS