PERFECT SINGLETON

Designing perfect singleton class:-

```java
package com.nit.perfect;

public class Printer {

    public static Printer instance;
    private static boolean flag=false;

    //0 param constructor
    //strict constructor for reflection api
    private Printer() {
        if(flag==true) {
            throw new IllegalArgumentException("object alredy created");
        }
        flag=true;
        System.out.println("0 param constructor");
    }

    //static factory method
    public static  Printer getInstance() {
        if(instance==null) {
            synchronized (Printer.class) {   //class level locking
                if(instance==null) {         //double null check
                    instance=new Printer();
                }
            }
        }
        return instance;
    }

    //override clone method from object
    @Override
    public Object clone() throws CloneNotSupportedException {
        System.out.println("Printer.clone()");
        return super.clone();
    }

    //by using deserialization
    public Object readResolve() {
        System.out.println("Printer.readResolve()");
        throw new IllegalArgumentException("object alredy created");
    }
}
```

PERFECT SINGLETON