

After refactoring:

Java program implementing Singleton class

// with method name as that of class

class Singleton

{

    // static variable single\_instance of type Singleton

    private static Singleton single\_instance=null;

    // variable of type String

    public String s;

    // private constructor restricted to this class itself

    private Singleton()

    {

        s = "Hello I am a string part of Singleton class";

    }

    // static method to create instance of Singleton class

    public static Singleton Singleton()

    {

        // To ensure only one instance is created

        if (single\_instance == null)

        {

            single\_instance = new Singleton();

        }

        return single\_instance;

    }

}

// Driver Code

class Main

{

    public static void main(String args[])

    {

        // instantiating Singleton class with variable x

        Singleton x = Singleton.Singleton();

        // instantiating Singleton class with variable y

        Singleton y = Singleton.Singleton();

        // instantiating Singleton class with variable z

        Singleton z = Singleton.Singleton();

        // changing variable of instance x

        x.s = (x.s).toUpperCase();

        System.out.println("String from x is " + x.s);

        System.out.println("String from y is " + y.s);

        System.out.println("String from z is " + z.s);

        System.out.println("\n");

        // changing variable of instance x

        z.s = (z.s).toLowerCase();

        System.out.println("String from x is " + x.s);

        System.out.println("String from y is " + y.s);

        System.out.println("String from z is " + z.s);

    }

}