Singleton Class in Java

In object-oriented programming, a singleton class is a class that can have only one object (an instance of the class) at a time.
After first time, if we try to instantiate the Singleton class, the new variable also points to the first instance created. So whatever modifications we do to any variable inside the class through any instance, it affects the variable of the single instance created and is visible if we access that variable through any variable of that class type defined.
To design a singleton class:

1. Make constructor as private.
2. Write a static method that has return type object of this singleton class..

```
class Singleton
{

    private static Singleton single_instance = null;


    public String s;


    private Singleton()
    {
        s = "Hello I am a string";
    }


    public static Singleton getInstance()
    {
        if (single_instance == null)
            single_instance = new Singleton();

        return single_instance;
    }
}


public class Practice
{
    public static void main(String[] args) {

        Singleton x = Singleton.getInstance();
```

```
        Singleton y = Singleton.getInstance();


        Singleton z = Singleton.getInstance();

        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");


        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);
    }
}
```

```
/Users/mohan/Library/Java/JavaVirtualMachines/openjdk-14
String from x is HELLO I AM A STRING
String from y is HELLO I AM A STRING
String from z is HELLO I AM A STRING


String from x is hello i am a string
String from y is hello i am a string
String from z is hello i am a string

Process finished with exit code 0
```

Double checked locking of Singleton

Double checked locking of Singleton is a way to ensure only one instance of Singleton class is created through an application life cycle. As the name suggests, in double-checked locking, code checks for an existing instance of Singleton class twice with and without locking to double ensure that no more than one instance of singleton gets created.

```
public static Singleton getInstanceDC() {
    if (single_instance == null) {
        synchronized (Singleton.class) {
            if (single_instance == null) {
                single_instance = new Singleton();
            }
        }
    }
```