```java
package lesson11.lecture.threadssingleton;

public class Singleton {
	private static Singleton instance;
	public static int counter = 0;
	private Singleton() {
		incrementCounter();
	}
	public static Singleton getInstance()
{
		if(instance == null) {
			instance = new
Singleton();
		}
		return instance;
	}
	private static void incrementCounter()
{
		counter++;
	}
}
```

```java
package lesson11.lecture.threadssingleton;

public class Singleton2 {
	private static Singleton2 instance;
	public static int counter = 0;
	private Singleton2() {
		incrementCounter();
	}
	public static Singleton2 getInstance() {
		if(instance == null) {
			instance = new Singleton2();

		}
		return instance;
	}
	/* Guarantees proper count of instances */
	synchronized private static void
incrementCounter() {
		counter++;
	}
}
```

```java
package lesson11.lecture.threadssingleton;


public class SingleThreadedTest2 {
	public static void main(String[] args)
{
		for(int i = 0; i < 1; ++i) {
			createAndStartThread();
			System.out.println("Num
instances: " + Singleton.counter);

		}
	}
	public static void
createAndStartThread() {
		Runnable r = () -> {
			for(int i = 0; i < 1000;
++i) {

	Singleton.getInstance();
			}
		};
		new Thread(r).start();
		try {
			Thread.sleep(10);
		} catch(InterruptedException e)
{}
	}
}
```

```java
package lesson11.lecture.threadssingleton;

public class SynchronizedSingleton {
	private static SynchronizedSingleton
instance;
	public static int counter = 0;
	private SynchronizedSingleton() {
		incrementCounter();
	}
	synchronized public static
SynchronizedSingleton getInstance() {
		if(instance == null) {
			instance = new
SynchronizedSingleton();
		}
		return instance;
	}
	private static void incrementCounter() {
		counter++;
	}
}
```