Exercise 1: Implementing the Singleton Pattern

Logger.java :

```
package singletonPatternExample;
public class Logger {
	private static Logger logger;
	private Logger() {
	}
	public static Logger getLogger() {
		if(logger==null) {
			logger = new Logger();
		}
		return logger;
	}
	public void display(String message) {
		System.out.println(message);
	}
}
```

Test.java :

```
package singletonPatternExample;
public class Test {
	public static void main(String[] args) {
		Logger logger1 = Logger.getLogger();
		Logger logger2 = Logger.getLogger();
		logger1.display("First message");
		logger2.display("Second message");
		if (logger1 == logger2) {
		    System.out.println("Both logger instances are the same. Singleton
verified.");
```

```
            }else {
                System.out.println("Different instances exist. Singleton failed.");
        }
          }
}
```

```
1  package singletonPatternExample;
2
3  public class Test {
4
5      public static void main(String[] args) {
6          Logger logger1 = Logger.getLogger();
7          Logger logger2 = Logger.getLogger();
8
9          logger1.display("First message");
10         logger2.display("Second message");
11
12         if (logger1 == logger2) {
13             System.out.println("Both logger instances are the same. Singleton verified
14         } else {
15             System.out.println("Different instances exist. Singleton failed.");
16         }
17
18     }
19
20 }
21
```

Console Coverage

<terminated> Test (1) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (22-Jun-2025, 2:41:15 pm – 2:41:15 pm) [pid: 180

```
First message
Second message
Both logger instances are the same. Singleton verified.
```