Exercise 1: Implementing the Singleton Pattern

Logger.java:

```
public class Logger {
    private static Logger instance;
    private Logger() {
        System.out.println("Logger Initialized");
    }
    public static Logger getInstance() {
        if (instance==null) {
            instance=new Logger();
        }
        return instance;
    }
    public void log(String message) {
        System.out.println("[LOG]: "+message);
    }
}
```

TestLogger.java:

```
public class TestLogger {
    public static void main(String[] args) {
        Logger logger1=Logger.getInstance();
        Logger logger2=Logger.getInstance();

        logger1.log("Message:Logger 1");
        logger2.log("Message:Logger 2");

        if (logger1==logger2) {
            System.out.println("Only one instance of Logger exists");
        } else {
            System.out.println("Multiple instances of Logger exist");
        }
    }
}
```

Output:

```
[Running] cd "/Users/yagnanarayanan/Library/Mobile Documents/com~apple~TextEdit/Documents/week1-cog/week1/Design Patterns and Principles/
SingletonPatternExample/" && javac TestLogger.java && java TestLogger
Logger Initialized
[LOG]: Message:Logger 1
[LOG]: Message:Logger 2
Only one instance of Logger exists

[Done] exited with code=0 in 0.505 seconds
```