

EXERCISE 1:

IMPLEMENTING SINGLETON PATTERN

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

```
public class LoggerTest {  
    public static void main(String[] args) {  
        Logger logger1 = Logger.getInstance();  
        logger1.log("First log message.");  
  
        Logger logger2 = Logger.getInstance();  
        logger2.log("Second log message.");  
  
        if (logger1 == logger2) {  
            System.out.println("Both logger1 and logger2  
are the same instance.");  
        } else {  
            System.out.println("Different instances  
created! Singleton failed.");  
        }  
    }  
}
```

USING JDoodle COMPILER



The screenshot shows the JDoodle compiler interface. At the top, there are two tabs: "Output" and "Generated files". The "Output" tab is selected, displaying the following text:

```
Logger instance created.  
Log: First log message.  
Log: Second log message.  
Both logger1 and logger2 are the same instance.
```

At the bottom of the interface, there is a status bar that reads: "Compiled and executed in 1.939 sec(s)".