Step 1: Create a New Java Project
Project Name: `SingletonPattern`

Step 2 & 3: Define and Implement Singleton Class

File: Logger.java

```
public class Logger {
    // Private static instance of the same class
    private static Logger instance;

    // Private constructor to restrict instantiation
    private Logger() {
        System.out.println("Logger Initialized");
    }

    // Public method to provide access to the instance
    public static Logger getInstance() {
        if (instance == null) {
            instance = new Logger();  // Lazy initialization
        }
        return instance;
    }

    // Logging method
    public void log(String message) {
        System.out.println("[LOG]: " + message);
    }
}
```

Step 4: Test the Singleton Implementation

File: Main.java

```
public class Main {
    public static void main(String[] args) {
        Logger logger1 = Logger.getInstance();
        logger1.log("First log message");

        Logger logger2 = Logger.getInstance();
        logger2.log("Second log message");

        // Check if both logger1 and logger2 refer to the same instance
        if (logger1 == logger2) {
            System.out.println("Both logger1 and logger2 are the same instance.");
        } else {
            System.out.println("Different logger instances exist.");
        }
    }
}
```

Expected Output:
Logger Initialized
[LOG]: First log message
[LOG]: Second log message
Both logger1 and logger2 are the same instance.

Final Output: