|  |
| --- |
| Exercise 1: Implementing the Singleton Pattern |

public class Logger {

    private static Logger instance;

    private Logger() {}

    public static Logger getInstance() {

        if (instance == null) {

            instance = new Logger();

        }

        return instance;

    }

    public void log(String message) {

        System.out.println("Log: " + message);

    }

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        Logger logger2 = Logger.getInstance();

        logger1.log("This is the first log message.");

        logger2.log("This is the second log message.");

        if (logger1 == logger2) {

            System.out.println("\nBoth logger1 and logger2 are the same instance.");

        } else {

            System.out.println("\nlogger1 and logger2 are different instances.");

        }

    }

}

**OUTPUT**

[Running] cd "c:\Users\rishi\" && javac Logger.java && java Logger

Log: This is the first log message.

Log: This is the second log message.

Both logger1 and logger2 are the same instance.

[Done] exited with code=0 in 2.437 seconds