**Exercise 1: Implementing the Singleton Pattern**

**CODE:**

// Singleton class for Logger

class Logger {

    // Holds the single instance

    private static Logger instance;

    // Private constructor prevents instantiation

    private Logger() {

        System.out.println("Logger Initialized.");

    }

    // Public method to get the only instance

    public static Logger getInstance() {

        if (instance == null) {

            instance = new Logger();

        }

        return instance;

    }

    // Simulate writing a log message

    public void log(String message) {

        System.out.println("[LOG]: " + message);

    }

}

// Client class

public class E {

    public static void main(String[] args) {

        // Get Logger instance and log some events

        Logger logger1 = Logger.getInstance();

        logger1.log("Application started.");

        logger1.log("User clicked login.");

        // Get Logger again — still same instance

        Logger logger2 = Logger.getInstance();

        logger2.log("Database connection opened.");

        logger2.log("Data retrieved successfully.");

        // Check if both logger instances are the same

        if (logger1 == logger2) {

            System.out.println("Both logger instances are the same (Singleton).");

        } else {

            System.out.println("Different instances (Not a Singleton).");

        }

    }

}

**OUTPUT:**

