EXERCISE-1: SINGLETON PATTERN EXAMPLE

**Singleton Pattern:**

The Singleton Pattern is a creational design pattern that ensures a class has only one instance and provides a global point of access to that instance.

**Logger.java:**

public 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);

    }

}

**TestLogger.java:**

public class TestLogger {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        Logger logger2 = Logger.getInstance();

        logger1.log("First message.");

        logger2.log("Second message.");

        if (logger1 == logger2) {

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

        } else {

            System.out.println("Singleton failed. Different instances found.");

        }

    }

}