**Exercise 1: Implementing the Singleton Pattern**

Logger.java – Singleton Class

public class Logger {

// Step 1: Create a private static instance of Logger

private static Logger instance;

// Step 2: Make constructor private to prevent external instantiation

private Logger() {

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

}

// Step 3: Provide a public static method to get the instance

public static Logger getInstance() {

if (instance == null) {

instance = new Logger(); // Create instance if it doesn't exist

}

return instance;

}

// Logging method

public void log(String message) {

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

}

}

LoggerTest.java – Testing Singleton Implementation

public class LoggerTest {

public static void main(String[] args) {

// Getting Logger instances

Logger logger1 = Logger.getInstance();

Logger logger2 = Logger.getInstance();

// Logging messages

logger1.log("First message");

logger2.log("Second message");

// Verifying both instances are the same

if (logger1 == logger2) {

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

} else {

System.out.println("Logger instances are different (Singleton failed).");

}

}

}