**DESIGN PRINCIPLES & PATTERNS**

# **Exercise 1: Implementing the Singleton Pattern**

## **Code**:

### ***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: " + message);  } } |
| --- |

### ***Loggertest.java:***

| public class Loggertest {  public static void main(String[] args) {   Logger log1 = Logger.getInstance();  Logger log2 = Logger.getInstance();   log1.log("This is the first message.");  log2.log("This is the second message.");   if (log1 == log2) {  System.out.println("Both are the same Logger instance.");  } else {  System.out.println("Different instances found!");  }  } } |
| --- |

## **Output Screenshot**:

