**Exercise 1: Implementing the Singleton Pattern**

**Scenario:**

You need to ensure that a logging utility class in your application has only one instance throughout the application lifecycle to ensure consistent logging.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **SingletonPatternExample**.
2. **Define a Singleton Class:**
   * Create a class named Logger that has a private static instance of itself.
   * Ensure the constructor of Logger is private.
   * Provide a public static method to get the instance of the Logger class.
3. **Implement the Singleton Pattern:**
   * Write code to ensure that the Logger class follows the Singleton design pattern.
4. **Test the Singleton Implementation:**
   * Create a test class to verify that only one instance of Logger is created and used across the application.

**Solution:**

1. At first created the Java Project in Eclipse IDE with the name “SingletonPatternExample”
2. Created Logger.java class
3. Created Main.java Class

**Source Code for Main.java:**

package SingletonPatternExample;

public class Main {

public static void main(String[] args) {

Logger logger1 = Logger.*getInstance*();

logger1.log("Application started.");

Logger logger2 = Logger.*getInstance*();

logger2.log("Performing Tasks..");

// Checking if both instances are the same

if (logger1 == logger2) {

System.***out***.println("logger1 and logger2 are both the same instance.");

} else {

System.***out***.println("Different instances not in Singleton.");

}

}

}

**Source code for Logger.java:**

package SingletonPatternExample;

public class Logger {

//private static instance

private static Logger *instance* = null;

//private constructor to restrict instantiation

private Logger() {

System.***out***.println("Logger instance created.");

}

//method to provide access to the instance

public static Logger getInstance() {

if (*instance* == null) {

*instance* = new Logger();

}

return *instance*;

}

//logging method

public void log(String message) {

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

}

}

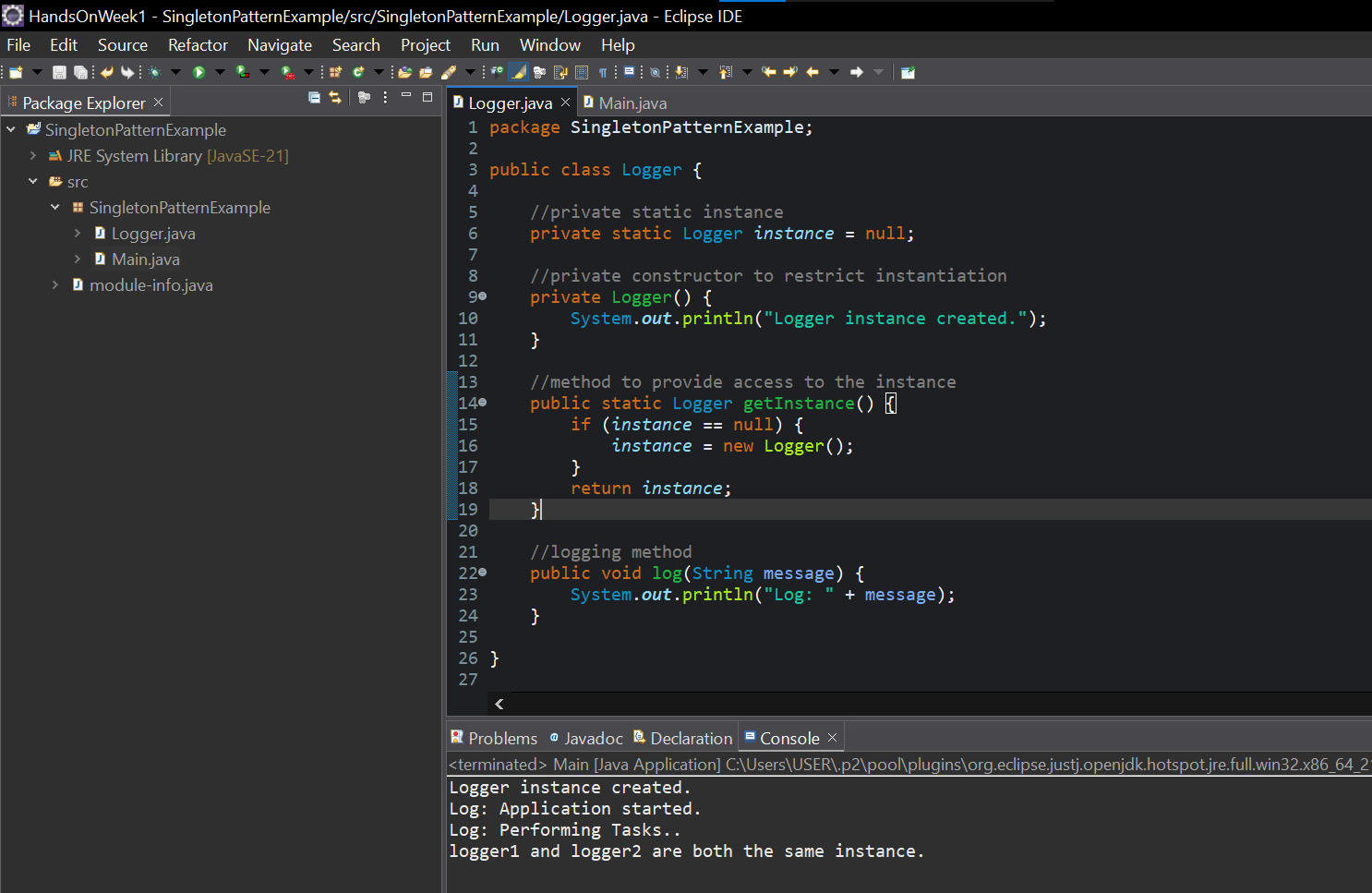
**Output:**

Logger instance created.

Log: Application started.

Log: Performing Tasks..

logger1 and logger2 are both the same instance.

**Screenshot of Eclipse IDE Application:**