**Step 1: Create a Java Project**

* **Project Name:** SingletonPatternExample
* You can do this in any IDE like **VS Code**, **Eclipse**, or just by creating a folder.

**Step 2: Define a Singleton Class**

public class SingletonPatternExample {

static class Logger {

private static Logger instance;

private Logger() {

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

}

public static Logger getInstance() {

if (instance == null) {

instance = new Logger();

}

return instance;

}

public void log(String message) {

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

}

}

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

logger1.log("First log");

Logger logger2 = Logger.getInstance();

logger2.log("Second log");

if (logger1 == logger2) {

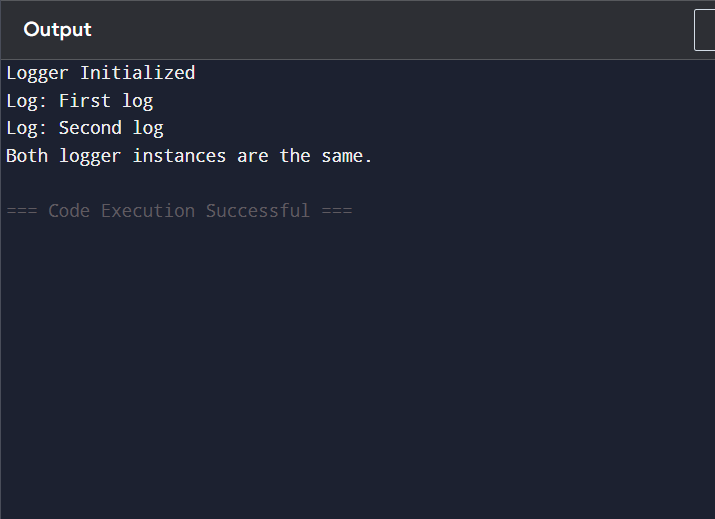
System.out.println("Both logger instances are the same.");

} else {

System.out.println("Different logger instances (should not happen).");

}

}

}

**Step 3: Test Singleton Pattern**

Create a separate test class:

java

CopyEdit

public class Main {

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

logger1.log("First log");

Logger logger2 = Logger.getInstance();

logger2.log("Second log");

// Check if both references are the same

if (logger1 == logger2) {

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

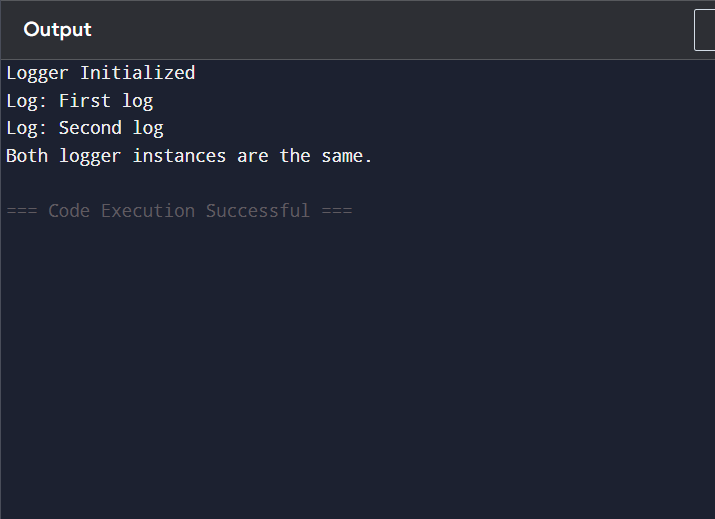
} else {

System.out.println("Different logger instances (should not happen).");

}

}

}



**💡**

**Notes:**

* This is a **lazy-initialized singleton**.
* Thread-safe version can use synchronized or double-checked locking (for multithreaded apps).