**Design Patterns and Principles**

**Exercise 1:**

**Implement a Singleton Pattern:**

Code:  
Logger.java

**package** singleton;

**public** **class** Logger {

**private** **static** Logger *singleInstance*;

**private** Logger() {

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

}**public** **static** Logger getInstance() {

**if** (*singleInstance* == **null**) {

*singleInstance* = **new** Logger();

}

**return** *singleInstance*;

}**public** **void** log(String message) {

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

}

}

LoggerTest.java:

**package** singleton;

**public** **class** LoggerTest {

**public** **static** **void** main(String[] args) {

Logger logger1 = Logger.*getInstance*();

logger1.log("Starting the application...");

Logger logger2 = Logger.*getInstance*();

logger2.log("Performing some operations...");

**if** (logger1 == logger2) {

System.***out***.println("✅ Only one Logger instance exists. Singleton is working!

} **else** {

System.***out***.println("❌ Different instances created. Singleton failed.");

}

}

}

**Output:**