**WEEK -1**  
**DESIGN PATTERNS AND PRINCIPLES  
Exercise 1**: Implementing the Singleton Pattern  
**CODE:**

**PROJECT NAME:** SingletonPatternExample  
**CLASS NAME:** Logger  
Logger.java:

public 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);

}

}

**CLASS NAME:** LoggerTest  
LoggerTest.java:  
public class LoggerTest {

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

logger1.log("First log message");

Logger logger2 = Logger.getInstance();

logger2.log("Second log message");

if (logger1 == logger2) {

System.out.println("Only one Logger instance exists. Singleton works!");

}

else {

System.out.println("Multiple instances found. Singleton failed!");

}

}

}