**Week 1**

**Design principles & Patterns :-**

**Hands-on :-**

**Exercise 1: Implementing the Singleton Pattern**

Code : -

1. Logger.java

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);  
 }  
}

1. Test.java

public class test {

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("Both logger1 and logger2 are the same instance.");

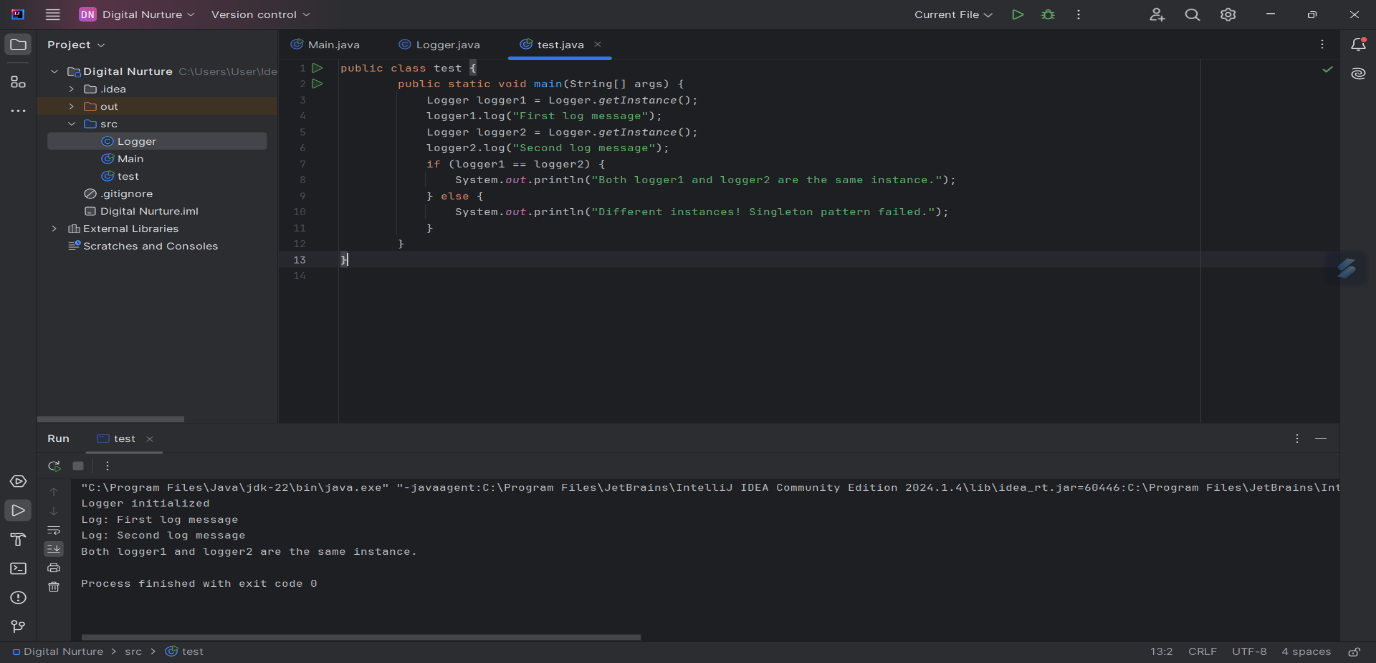
} else {

System.out.println("Both follows different instances! Singleton pattern failed.");

}

}

}

**OUTPUT**