**WEEK -1**

**Design Patterns and Principles**

**Exercise 1 : Implementing the Singleton Pattern**

**Singleton Project module**

Logger.java:

public class Logger {  
 private Logger(){  
 System.out.println("Logger Initialized");  
 }  
 private static class loggerHolder{  
 private static final Logger INSTANCE = new Logger();  
  
 }  
 public static Logger getInstance(){  
 return loggerHolder.INSTANCE;  
 }  
 public void logInfo(String message){  
 System.out.println("Info :"+message);  
 }  
  
}

Main.java:

public class Main {  
 public static void main(String[] args) {  
 Logger logger1 = Logger.*getInstance*();  
 Logger logger2 = Logger.*getInstance*();  
 logger1.logInfo("My application is started.");  
 logger2.logInfo("Second message");  
 if (logger1 == logger2) {  
 System.*out*.println("Singleton confirmed : Only one instance is used");  
  
 } else {  
 System.*out*.println("Singleton failed : multiple found");  
  
 }  
 }  
  
}

**Output For Singleton :**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**