**Design Patterns and Principles**

**Exercise 1: Implementing the Singleton Pattern**

**Scenario:**

You need to ensure that a logging utility class in your application has only one instance throughout the application lifecycle to ensure consistent logging

**Code:**

**Logger.java**

package SingletonPatternExamplee;

public class Logger {

private static Logger Instance ;

private Logger() {

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

}

public static Logger getInstance() {

if(Instance==null)

Instance =new Logger() ;

return Instance ;

}

}

**Main.java**

package SingletonPatternExamplee;

public class Main {

public static void main(String[] args) {

Logger log1=Logger.getInstance();

Logger log2=Logger.getInstance();

if(log1==log2) System.out.println("Singleton created") ;

else System.out.println("Failed") ;

}

}

**OUTPUT**:



