Design principles & Patterns

**Exercise 1**: Implementing the Singleton Pattern

**Logger.java**  
  
package com.singleton.example;

public class Logger {

private static Logger *instance*;

private Logger()

{

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

}

public static Logger getInstance()

{

if(*instance*==null)

{

*instance*=new Logger();

}

return *instance*;

}

}

**Main.java**  
package com.singleton.example;

public class Main {

public static void main(String[] args) {

Logger l1=Logger.*getInstance*();

Logger l2=Logger.*getInstance*();

Logger l3=Logger.*getInstance*();

if(l1==l2)

System.*out*.println("Same Instance");

else

System.*out*.println("Different Instance");

if(l1==l3)

System.*out*.println("Same Instance");

else

System.*out*.println("Different Instance");

if(l2==l3)

System.*out*.println("Same Instance");

else

System.*out*.println("Different Instance");

}

}

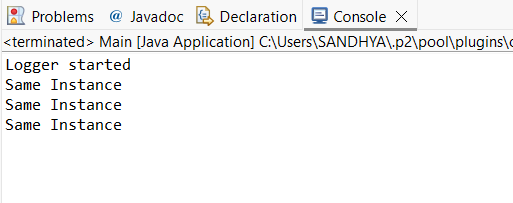
**OUTPUT**

Logger started

Same Instance

Same Instance

Same Instance

****