Exercise 1: Singleton Pattern for Logger Approach and Understanding:

In this exercise, I made a **Logger class that follows the Singleton Pattern.**

# The question wanted me to create a logging system that:

* Has **only one Logger object** in the entire application.

# Makes sure the same Logger is used everywhere to keep logs consistent.

**Code**

import java.util.Scanner; class L {

private static L one = null ; private L ( ) { }

public static L get ( ) { if ( one == null ) {

one = new L ( ) ;

}

return one ;

}

public void write ( String m ) { System . out . println ( m ) ;

}

}

public class X {

public static void main ( String [ ] z ) { Scanner sc = new Scanner ( System . in ) ;

L a = L . get ( ) ;

L b = L . get ( ) ;

System . out . print ( "Say 1: " ) ; String t1 = sc . nextLine ( ) ;

System . out . print ( "Say 2: " ) ; String t2 = sc . nextLine ( ) ;

1. write ( t1 ) ;
2. write ( t2 ) ;

if ( a == b ) {

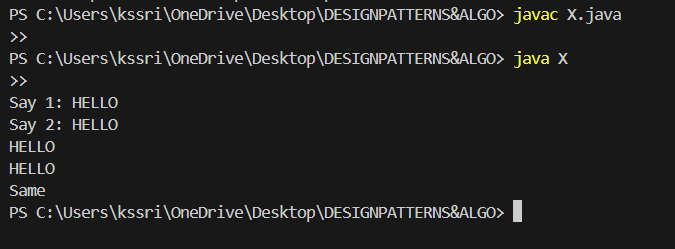
System . out . println ( "Same" ) ;

} else {

System . out . println ( "Not same" ) ;

}

sc . close ( ) ;



}

**OUTPUT:**