**Assisted Practice: 3.3 Thread Synchronization Mechanisms**

This section will guide you to:

* Write a program in Java to demonstrate synchronization
* Use Eclipse (the popular text editor for Java programs)
* Push code to Git

This lab has three sub-sections, namely:

* + 1. Creating a new project in Eclipse
    2. Writing a program in Java to demonstrate synchronization
    3. Pushing the code to your GitHub repositories

**Step 3.3.1:** Creating a new project in Eclipse

* Open Eclipse
* Go to File -> New -> Project -> Java Project -> Next.
* Type in any project name and click on “Finish.”
* Select your project and go to File -> New -> Class.
* Enter **SyncDemo** in any class name, check the checkbox “public static void main(String[] args)”, and click on “Finish.”

**Step 3.3.2:** Writing a program in Java to demonstrate synchronization

import java.io.\*;

import java.util.\*;

class Sender

{

public void send(String msg)

{

System.out.println("Sending\t" + msg );

try

{

Thread.sleep(1000);

}

catch (Exception e)

{

System.out.println("Thread interrupted.");

}

System.out.println("\n" + msg + "Sent");

}

}

class ThreadedSend extends Thread

{

private String msg;

private Thread t;

Sender sender;

ThreadedSend(String m, Sender obj)

{

msg = m;

sender = obj;

}

public void run()

{

synchronized(sender)

{

sender.send(msg);

}

}

}

class SyncDemo

{

public static void main(String args[])

{

Sender snd = new Sender();

ThreadedSend S1 =

new ThreadedSend( " Hlo " , snd );

ThreadedSend S2 =

new ThreadedSend( " Lakshmi " , snd );

S1.start();

S2.start();

try

{

S1.join();

S2.join();

}

catch(Exception e)

{

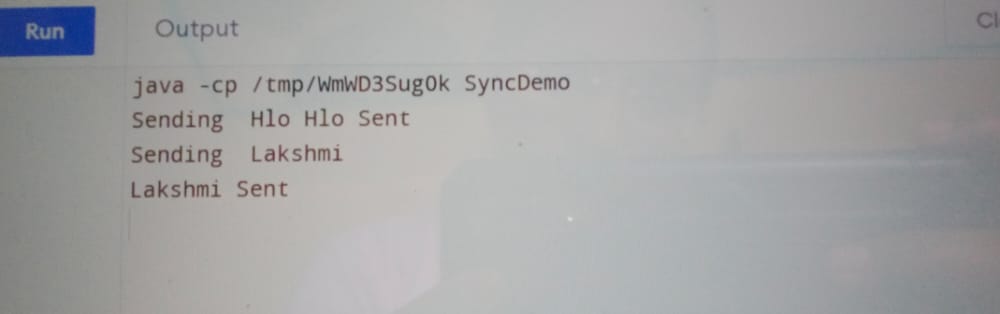
System.out.println("Interrupted");

}

}

}

**Output:**

****