Name: Ariyan Hossain

ID: 20101099

Sec: 08

**CSE 321 Lab 5:**

**Task 1:**

import java.util.ArrayList;

import java.util.List;

class QuestionBuffer {

public static int pointer = -1;

private List<String> registers = new ArrayList<>();

public synchronized String readQuestionFromReg() throws InterruptedException {

if (pointer==-1){

wait();

}

String a=registers.get(0);

pointer--;

registers.remove(0);

return a;

}

public synchronized void writeQuestionToReg(String value) {

registers.add(value);

pointer++;

if (pointer==0){

notifyAll();

}

}

}

// Do not modify this class

class TeacherThread extends Thread {

private String[] values = {

"What is your name?",

"What is your student ID?",

"Please mention your course title,theory section and lab section.",

"Have you received your first dose of covid-19 vaccine?",

"Have you received your second dose of covid-19 vaccine?",

"Are you prepared for offline classes in the upcoming semester?",

"Write a few lines to describe yourself.",

};

QuestionBuffer questionBuffer;

public TeacherThread(QuestionBuffer questionBuffer) {

this.questionBuffer = questionBuffer;

}

@Override

public void run() {

for (int i = 0 ; i < values.length ; i++) {

try {

questionBuffer.writeQuestionToReg(values[i]);

sleep((int)(Math.random() \* 1000));

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

// Do not modify this class

class StudentThread extends Thread {

QuestionBuffer questionBuffer;

public StudentThread(QuestionBuffer questionBuffer) {

this.questionBuffer = questionBuffer;

}

@Override

public void run() {

try {

while (true) {

System.out.println(Thread.currentThread().getName() + " prints: " + questionBuffer.readQuestionFromReg());

}

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

// Do not modify this class

public class LabTask{

public static void main(String[] args) throws InterruptedException {

QuestionBuffer questionBuffer = new QuestionBuffer();

StudentThread studentThread = new StudentThread(questionBuffer);

TeacherThread teacherThread = new TeacherThread(questionBuffer);

teacherThread.start();

studentThread.start();

teacherThread.join();

studentThread.stop();

System.out.println(QuestionBuffer.pointer);

}

}