COLLEGE OF ARTS AND SCIENCES
SCHOOL OF COMPUTING

STIQK3014
REAL-TIME PROGRAMMING

GROUP A

GROUP DISCUSSION

LECTURER:

DR. RUZITA BINTI AHMAD

PREPARED BY:

NUR SYAZALINA BINTI BADRUL HISHAM 297527
NUR ATIRAH BINTI MOHD RIDZUAN 300697
MUHAMMAD NABIL IKHWAN BIN NAZIRUDDIN 294428

STIWK3014 REAL TIME PROGRAMMING
Tutorial / Exercise 10: TrafficLightController by using ReentrantLock Method (Group Discussion)

Instruction:
Simulate a traffic light controller for a 4-way intersection where only one direction can have the
green light at a time. Each direction has its own thread controlling the light using Java's for synchronization. The concept used are:

- `ReentrantLock` from `java.util.concurrent.locks`
- Thread-based simulation (4 threads for 4 directions)
- Light states: `GREEN`, `YELLOW`, `RED`
- Fair scheduling using locks

Submission:

Platform: 1. Online Learning - Sample Coding & Output in PdF form
2. GitHub – Upload the coding file to your GitHub account.
3. Only leader will submit the answer.

Date: 21 May 2025 (Wednesday, before 12.30 noon)

Coding

```
import java.util.concurrent.locks.ReentrantLock;

public class TrafficLightController {

    private static final ReentrantLock lock = new ReentrantLock();
    private static final int GREEN_TIME = 5000;  // milliseconds
    private static final int YELLOW_TIME = 2000;

    public static void main(String[] args) {
        for (int i = 1; i <= 4; i++) {
            final int direction = i;
            new Thread(() -> controlLight(direction)).start();
            try {
                Thread.sleep(GREEN_TIME + YELLOW_TIME); // Stagger thread start
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }

    private static void controlLight(int direction) {
        while (true) {
            lock.lock();
            try {
                System.out.println("Direction " + direction + ": GREEN");
                Thread.sleep(GREEN_TIME);

                System.out.println("Direction " + direction + ": YELLOW");
                Thread.sleep(YELLOW_TIME);

                System.out.println("Direction " + direction + ": RED\n");
            } catch (InterruptedException e) {
                e.printStackTrace();
            } finally {
                lock.unlock();
            }

            // Sleep a bit before trying again
            try {
                Thread.sleep(1000); // avoid starvation
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

pom.xml × TrafficLightController.java ×

```java
import java.util.concurrent.locks.ReentrantLock;

public class TrafficLightController {

    private static final ReentrantLock lock = new ReentrantLock();  2 usages
    private static final int GREEN_TIME = 5000;  // milliseconds  2 usages
    private static final int YELLOW_TIME = 2000;  2 usages

    public static void main(String[] args) {
        for (int i = 1; i <= 4; i++) {
            final int direction = i;
            new Thread(() -> controlLight(direction)).start();
            try {
                Thread.sleep( millis: GREEN_TIME + YELLOW_TIME); // Stagger thread start
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }

    private static void controlLight(int direction) {  1 usage
        while (true) {
            lock.lock();
            try {
                System.out.println("Direction " + direction + ": GREEN");
                Thread.sleep(GREEN_TIME);

                System.out.println("Direction " + direction + ": YELLOW");
                Thread.sleep(YELLOW_TIME);

                System.out.println("Direction " + direction + ": RED\n");
            } catch (InterruptedException e) {
                e.printStackTrace();
            } finally {
                lock.unlock();
            }

            // Sleep a bit before trying again
            try {
                Thread.sleep( millis: 1000); // avoid starvation
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Output

```
Run    TrafficLightController
"C:\Program Files\Java\openjdk-24_windows-x64_bin\jdk-2
Direction 1: GREEN
Direction 1: YELLOW
Direction 1: RED

Direction 2: GREEN
Direction 2: YELLOW
Direction 2: RED

Direction 1: GREEN
Direction 1: YELLOW
Direction 1: RED

Direction 3: GREEN
Direction 3: YELLOW
Direction 3: RED

Direction 2: GREEN
Direction 2: YELLOW
Direction 2: RED

Direction 4: GREEN
Direction 4: YELLOW
Direction 4: RED
```