Enterprise Architecture – 4232

Lab sheet 02

Task 03

3.**counter class**

public class Counter {

private int count = 0;

// Synchronized method to ensure thread-safe access to the counter

public synchronized void increment() {

count++;

}

public int getCount() {

return count;

}}

**SynchronizedExample class**

**public class SynchronizedExample extends Thread{**

**private Counter counter;**

**public SynchronizedExample(Counter counter) {**

**this.counter = counter;**

**}**

**@Override**

**public void run() {**

**for (int i = 0; i < 1000; i++) {**

**counter.increment();**

**}**

**}**

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

**Counter counter = new Counter();**

**// Create and start multiple threads**

**Thread thread1 = new SynchronizedExample(counter);**

**Thread thread2 = new SynchronizedExample(counter);**

**thread1.start();**

**thread2.start();**

**// Wait for threads to finish**

**thread1.join();**

**thread2.join();**

**System.out.println("Final counter value: " + counter.getCount());**

**}}**

Output

