Assignment14.java

```java
public class Assignment14 {
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("One command line parameter only: must be an interger
(or long) number");
        } else {
            long iterationsPerThread = Long.parseLong(args[0]);

            // Call to student's class:
            // takes number of iterations per thread as input
            // returns result of two threads each incrementing a shared
            // variable "in" (initialised with 0) "iterationsPerThread" times.
            long result = MyAssignment14.main(iterationsPerThread);

            System.out.println(result);
        }
    }
}
```

MyAssignment14.java

```java
import java.util.concurrent.Semaphore;

public class MyAssignment14 extends Thread { // TODO: Modify as necessary

    private static Counter counter;
    private static long max;
    private static Semaphore sem;

    public static long main(long iterationsPerThread) { // Do not modify this line!
        Thread thread1 = new MyAssignment14();
        Thread thread2 = new MyAssignment14();

        thread1.setName("0");
        thread2.setName("1");

        sem = new Semaphore(1);

        max = iterationsPerThread;
        counter = new Counter();

        thread1.start();
        thread2.start();

        try {
            thread1.join();
            thread2.join();
        }
        catch(Exception ex) {
            System.out.println("Exception" + ex);
        }
        return counter.getIn();
    }

    public void run() {
        try {
            for (int i = 1; i <= max; i++) {
                System.err.print(this.getName());
                sem.acquire();
                counter.increment(max);
                sem.release();
            }
        } catch (InterruptedException ie) {
            //error
        }
    }
}
```

Counter.java

```java
public class Counter {
    public static volatile long in = 0;
    public void increment(long max) {
        long next_free_slot = in + 1;
        in = next_free_slot;
    }

    public long getIn() {
        return in;
    }
}
```