MyAssignment12.java

```
public class Assignment12 {
    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 = MyAssignment12.main(iterationsPerThread);

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

MyAssignment12.java

```
public class MyAssignment12 extends Thread {
    public static volatile long in;
    private static long myIteration;
    private int id;
    public static volatile boolean[] flag = {false,false};
    public static volatile int turn;

    public MyAssignment12 (long iterationsPerThread,int id) {
        myIteration = iterationsPerThread;
        in = 0;
        this.id = id;
    }

    public static long main(long iterationsPerThread) { // Do not modify this line!
        Thread eitt = new MyAssignment12(iterationsPerThread,0);
        Thread tvo = new MyAssignment12(iterationsPerThread,1);

        eitt.start();
        tvo.start();

        try {
            eitt.join();
            tvo.join();
        }
        catch (InterruptedException ie) {
```

```
            System.out.println("Interrupted while waiting thread");
        }

        return in;
    }

    public void run() {
        for (int i = 1; i <= myIteration; i++){
            increment();
        }
    }

    public void increment() {
        flag[id] = true;
        turn = 1-id;
        while(flag[1-id] && turn==1-id) {}
        long next_free_slot = in;
        next_free_slot ++;
        in = next_free_slot;
        flag[id] = false;
    }
}
```