Week 4 Exercises

1. List 4 benefits of multithreading and explain why each is possible.

2. A computer game, controlled by the keyboard, has a character that the user must control, a single 'enemy' that will chase the user, suggest 5 different threads this program could use to parallelise its activity.

3. Why might a web server need to be multithreaded?

4. Explain what it means for an instruction to be atomic, and why incrementing the value of a counter is not an atomic instruction.

5. Explain why the sharing of an array between two processes that add data to it might lead to information loss.

6. Consider the following solution to the critical section problem (Attempt 1)
   a) Show that this solution satisfies mutual exclusion without using a state diagram (i.e. the way we did for attempts 3 and 4, or by induction if you prefer).
   b) Which properties are not satisfied by this solution? Explain your answer.

integer turn ← 1	
p	q
loop_forever	loop_forever
p1: non-critical section	q1: non-critical section
p2: await turn = 1	q2: await turn = 2
p3: critical section	q3: critical section
p4: turn ← 2	q4: turn ← 1

7. Show mutual exclusion for the following solution to the critical section problem (Attempt 3) by drawing the state diagram. For brevity note that you can achieve the same result whilst only considering the lines relevant to syncrhonisation (numbered lines). Which properties are not satisfied by this solution? Explain your answer.

boolean wantp ← false, wantq ← false	
p	q
loop_forever	loop_forever
non-critical section	non-critical section
p1: wantp ← true	q1: wantq ← true
p2: await wantq = false	q2: await wantp = false
p3: critical section	q3: critical section
p3: wantp ← false	q4: wantq ← false

8. State whether the following solution to the critical section problem (Attempt 4) satisfies each of the 3 properties of a solution to the aforementioned problem. If the property is satisfied provide a proof (in the style of the lecture slides); if it is not show an interleaving that breaks that property.

boolean wantp ← false, wantq ← false	
p	q
loop_forever p1: non-critical section p2: wantp ← true P3: while wantq p4: wantp ← false P5: wantp ← true p6: critical section p7: wantp ← false	loop_forever q1: non-critical section q2: wantq ← true q3: while wantp q4: wantq ← false q5: wantq ← true q6: critical section q7: wantq ← false

9. The following code can be used to implement an atomic assignment in Java:

```
public class AtomicInt {

      int value;

      public AtomicInt(int initialValue){
        value = initialValue;
      }

      public synchronized void assign(int newValue){
            value = newValue;
      }
}
```

a) Create a similar class `AtomicBoolean` that provides the same functionality for a boolean variable.
b) By using Peterson's algorithm and the classes above**, and without using the keyword `Synchronized` in the buffer class**, create a buffer class, similar to the one in the lecture slides that allows threads to add ints to the buffer, you may assume that the buffer will be sufficiently large that it does not become full (do not implement remove). Hint: recall you can implement await using while and sleep.
c) Write a main method that:
   i. Creates an instance of your buffer object of size 10;
   ii. Creates two producer threads (you may use the classes from existing lecture code or write your own) that reference your buffer and loop 5 times adding the number 1 or 2 respectively to the buffer, printing the buffer after each addition.
   iii. Starts these threads running.
d) Test the output of your code (remember to make sure you put the waste some time code in the run() method of the threads in order to make it more likely that you will spot bugs.
e) You can extend this exercise by implementing the other three solutions in the lecture and observing the output.

10. Revision exercises: for each of the solutions to the Mutual Exclusion problem:
a) Write out the solution then close your notes;
b) State which of the properties the solution satisfies and provide a proof for each;
c) State which of the properties the solution does not satisfy and provide an example interleaving to demonstrate this.