CIS 720: Advanced Operating Systems, Fall 2013
Homework 1, Due October 1, 2013

1. You need to write a program to control traffic on a one lane bridge. At any given time there can be cars crossing in one direction only (left or right). However, there can be multiple cars in any one direction. Write an invariant, generate the await statement based solution, and then come up with a semaphore based solution. The skeleton code for the car threads is shown below.

```
Left_car()
        {
        do
                wait until it is safe to cross the bridge;
                cross the bridge from the left side
        od
        }
right_car()
        {
        do
                wait until it is safe to cross the bridge;
                cross the bridge from the right side
        od
        }
```

2. Consider the program S:

```
x := 0
co
 x := x + 2
||
< x = x – 3 >
||
< x := x + 4 >
oc
```

   Prove that { true } S { x = 3 } is true.

3. For each of the following, show whether the precondition and the statement:
   { x >= 4 } x := x – 4
   interferes with the following assertions:
   (a) x >= 0
   (b) x is even
   (c) x is odd
   (d) x < 0
   (e) y > 0

4. Develop a proof outline for the following program, prove the proof outline and use the proof outline to show that two processes cannot be in the critical section at the same time and the solution is deadlock free (Hint: use (1 - sem = in1 + in2) /\ sem <= 1 as the invariant).

```
sem = 1; in1 = 0; in2 = 0
co
   P1:
    do
      < await sem ==1 → sem = sem - 1; in1 = 1>
      use critical section
     <sem = sem + 1; in1 = 0 >
      non critical section
   od
   ||
   P2:
    do
      < await sem == 1 → sem = sem - 1; in2 = 1>
      use critical section
     <sem = sem + 1; in2 = 0 >
      non critical section
   od
oc
```