COP4020 Spring 2010 Homework Assignment 4

1. Consider the following tic-tac-toe board positions:

X	O	O
	X	X
O		

That is, the program database holds the following facts (in this order):

```
x(1)
x(5)
x(6)
o(2)
o(3)
o(7)
```

Show a trace of the execution of sub-goals to solve the goal '**`move(X).`**'.

Note: use the tic-tac-toe program shown in the lecture notes, not the one in the textbook, which is slightly different. You can download the program from:
http://www.cs.fsu.edu/~engelen/courses/COP4020/tictactoe.gz
On linprog, start Prolog with '**`pl`**' or '**`swipl`**' (or '**`xpce`**' when available) and load the program with '**`[tictactoe].`**' (don't forget the period to terminate the command). Type '**`move(X).`**' to query the system, but notice that the starting board positions are different in this demo. Use '**`listing.`**' to show the program database. Use '**`trace.`**' to activate the tracer before you enter a goal (use ? for help and type ENTER to creep

through the trace). The tracer will help you with this assignment, but in addition you have to write down the trace depth (indentation), show failures, and redos.

2. Now change the position **o(7)** into **o(9)** in the program and trace '**move(X).**'. You will notice it takes more steps to find the winning spot for the X.
Let's change the program to see if we can speed it up. Try changing the **move** clause definition into:

```
move(A) :- empty(A), good(A), !.
```

Try '**move(X).**'. What happens and why?

3. Fix the program so that the new '**move**' clause works. Hint: you might want to change the definition of '**empty**' so that it instantiates the variable to the position of an empty spot.

4. Textbook exercise 11.6 (2nd ed). Consider the Prolog program:

```
takes(jane_doe, his201).
takes(jane_doe, cs254).
takes(ajit_chandra, art302).
takes(ajit_chandra, cs254).
classmates(X, Y) :- takes(X, Z), takes(Y, Z).
```

The query

```
?- classmates(jane_doe, X).
```

will succeed three times: twice with **X = jane_doe** and once with **X = ajit_chandra**. Show how to modify the **classmates(X, Y)** rule so that a student is not considered a classmate of him or herself.

5. Take a small C program from any one of your CS class projects and compile it on linprog with:
$ gcc –v program.c
Which tools are invoked by gcc when you run this? What other files besides a.out are generated temporarily?

6. Textbook exercise 1.1 (2nd ed): Errors in a computer program can be classified according to when they are detected and, if they are detected at compile time, what part of the compiler detects them. Using your favorite imperative language [RvE: use C], give an example of each of the following.

a) A lexical error, detected by the scanner (hint: the scanner accepts only legal characters)

b) A syntax error, detected by the parser

c) A static semantics error, detected by semantic analysis

d) A dynamic semantic error, detected by the code generated by the compiler

e) An error that the compiler can neither catch not easily generate code to catch (this could be a violation of the language definition, not just a program bug).