CS 359 – Programming Paradigms

PRELIM 4 – PERT

Due: Start of Class, Lesson 35, 25 Nov

Help Policy:

AUTHORIZED RESOURCES: Any, except

1) another cadet's program.
2) **Any website containing the solutions to the exact problems listed below**

NOTE:

- Never copy another person's work and submit it as your own.
- Do not jointly create a program unless explicitly allowed.
- You must document all help received from sources other than your instructor or instructor-provided course materials (including your textbook).
- **DFCS will recommend a course grade of F for any cadet who egregiously violates this Help Policy or contributes to a violation by others.**

Documentation Policy:

- You must document all help received from any source other than your instructor.
- The documentation statement must explicitly describe WHAT assistance was provided, WHERE on the assignment the assistance was provided, and WHO provided the assistance.
- If no help was received on this assignment, the documentation statement must state "NONE."
- If you checked answers with anyone, you must document with whom on which problems. You must document whether or not you made any changes, and if you did make changes you must document the problems you changed and the reasons why.
- **Vague documentation statements must be corrected before the assignment will be graded and will result in a 5% deduction on the assignment.**

Turn-in Policies:

- On-time turn-in is at the specific time listed above.
- **No late submissions will be accepted for this PRELIM.**
- There is no early turn-in bonus or extra credit for this assignment.

1. OBJECTIVES

- Expand your range of programming language competence
- Be able to define and use Prolog facts and rules
- Be able to use the Prolog list structure
- Be able to create, test, and debug a medium sized Prolog program
- Be able to represent a PERT chart using Prolog and obtain relevant information from it

2. INSTRUCTIONS

a) Read the PEX4 assignment document thoroughly. Develop a set of questions whose answers would make you better able to complete PEX4. Ask those questions in class

b) Answer the Questions to Get You Thinking below

c) Implement the required Preliminary Exercise for PERT from section 4.

d) Follow submission requirements below.

e) DO NOT use any built-in Prolog predicates to solve any of the problems below. That includes length/2, reverse/2, and any other.

3. QUESTIONS TO GET YOU THINKING

1) (2 points each) Write a Prolog rule (or a set of Prolog rules) to implement each of the following. Hint 1: if Prolog abbreviates the list answer to your query, press the w key to see the whole list. Hint2: each rule must work only for the first returned value of X. If subsequent presses of the semicolon key produce incorrect answers, that will not count against you. Hint 3: hint 2 means you can count on Prolog attempting to match your rules in the order you declare them. Hint 4: There is no need to write header comments for these rules.

   a. Remove the Kth element from a list
   ```
   removeKthElement([66,77,88,99], 3, X).
   X = [66, 77, 99] .
   ```
   b. Split a list into two parts. The user will specify the length of the first part.
   ```
   split([1,2,3,4,5], 3, L1, L2).
   L1 = [1, 2, 3],
   L2 = [4, 5] .
   ```
   c. Get rid of consecutive duplicates of list elements.
   ```
   removeDups([a,a,a,b,b,b,c,c,c,c,c,a], X).
   X = [a, b, c, a] .
   ```

2) (2 points) Express the following First Order Predicate Calculus (FOPC) statement in English. Use clear English understandable by people who do not study Logic, Math or Computer Science. Do not use variables like X or Y in your sentence.

$$\forall X, Y, L\ speaksLanguage(X, L) \cap speaksLanguage(Y, L) \Rightarrow understands(X, Y) \cap understands(Y, X)$$

3) (1 point each) Using the FOPC clauses below, write a FOPC sentence for each given English statement. Use lowercase letters for constants. Use UPPERCASE letters for variables.

   occupation(P,O): "Person P has occupation O"
   customer(P1, P2): "Person P1 is a customer of P2"

a. Emily is either a surgeon or a lawyer
b. Joe is an actor, but he also holds another job
c. All surgeons are doctors
d. Joe does not have a lawyer (i.e. Joe is not a customer of any lawyer)
e. There is some lawyer who has only doctors as clients

4) (3 points) For the following set of facts, draw a trace diagram representing how Prolog will show the clause `uncleOrAunt(bob, matt)` is true. You may use the Prolog interpreter `trace` to check your work. However, you MAY have to draw a trace diagram without using a computer before this course ends. Do not erase any part of your diagram in order to show backtracking. Instead, upon need to backtrack, draw a new diagram. It might be helpful to use MS OneNote or MS Powerpoint for this problem, as it allows you to copy and paste your work. **For any call to sibling/2, assume that Prolog will check only if bob is the correct sibling.**

```
parent(tim, matt).
parent(shirley, matt).
parent(anne, tim).
parent(paul, tim).
parent(robert, shirley).
parent(rita, shirley).
parent(robert, bob).
parent(rita, bob).


sibling(S1, S2) :-
     parent(P, S1),
     parent(P, S2),
     S1 \= S2.

uncleOrAunt(UA, N) :-
     parent(P, N),
     sibling(P, UA).
```

5) (3 points) Use resolution and unification on the following facts and rules to show that Leroy must report to Spring Training.
   a. Using resolution, combine two rules to create a larger rule. Do not change variable names yet.
   b. Unify variable names with constants or other useful variable names.
   c. Cancel like terms to give the rule stating that Leroy must report to Spring Training.

$$reportToSpringTraining(BP) \subset ballPlayer(BP)$$

$$ballPlayer(P) \subset pitcher(P)$$

$$ballPlayer(C) \subset catcher(P)$$

$$catcher(leroy)$$

4. PRELIMINARY EXERCISE FOR PERT (11 POINTS)

For the preliminary exercise you will create a PERT chart, drawing a diagram and writing its representations in Prolog, and create a **`path/3`** rule. The PERT chart should be about twice the size of the sample in this document. It should have at least three possible paths between the start event and end event with one being the critical path. It should describe a project you are familiar with.

The **`path/3`** rule will be used to determine paths between events in your PERT charts. Specifically, given the sample PERT chart in the PEX4 main document, the rule should behave as follows:

```
?- path( a, e, [foundation, walls, ceiling, painting] ).
true ;
false.

?- path( a, e, Path ).
Path = [foundation, walls, ceiling, painting] ;
Path = [foundation, walls, electric, painting] ;
Path = [foundation, plumbing, painting] ;
false.

?- path( c, c, Path ).
Path = [] ;
false.
```

5. SUBMISSION REQUIREMENTS

- **INCLUDE AN APPROPRIATE DOCUMENTATION STATEMENT IN YOUR MOODLE SUBMISSION.**
- For the preliminary exercise, you must create and submit five separate files:
  - **`prelim4Diagram.pdf`** – This file will contain a your small PERT diagram
  - **`prelim4Diagram.pl`** – This file will contain your facts for your small PERT diagram
  - **`prelim4Rules.pl`** - This file will contain your path/3 rule.
  - **`listProblems.pl`** – This file will contain your rules to answer Question To Get You Thinking #1
  - **`writeOutProblems.pdf -`** This file will contain your answers to Questions To Get You Thinking #2-#5
- To create a PDF from a Powerpoint, click File->Save As->Save As Type-> PDF. From OneNote, File->Save As->Page->Select Format->PDF
- Create a folder on your system with everything you would like to be graded and *name this folder* with *your own last name*.
- Zip the entire folder to a file with the name **`Lastname.zip`**, using your own last name.
- Use the website to submit a **single zip file** containing everything you would like to be graded.

CS 359 – PEX 4 Prelim – Grade Sheet Name:_____

Criteria		Points Earned	Points Available
Question 1 (3 parts)			**6**
Question 2			**2**
Question 3 (5 parts)			**5**
Question 4			**3**
Question 5 (3 parts)			**3**
PERT diagrams and Prolog code			**3**
path/3 rule			**8**
	Subtotal:		**30**
Adjustments	**All code meets specified standards:**		**– 3**
	Vague/Missing Documentation:		**– 1.5**
	Submission Requirements Not Followed:		**– 2**
	Total w/adjustments:		

Comments from Instructor: