## **CSE 240 Homework 12, Fall 2015 (50 points)**

Due Saturday, November 21, 2015 at 11:59PM, plus one day grace period

### Introduction

The aim of this assignment is to make sure that you understand and are familiar with the concepts covered in the lectures. By the end of the assignment, you should have

* Understood the concepts of logic / declarative programming paradigm
* strong concepts of facts, rules, and questions.

**Reading**: Chapter 5 and lecture slides.

**Preparation**: Complete the multiple choice questions in the textbook exercise section. The answer keys can be found in the course Web site. These exercises can help you prepare for your weekly quiz and the exam. You are encourage to read the other exercise questions and make sure you understand these questions in the textbook exercise section, which can help you better understand what materials are expected to understand after the lectures and homework on each chapter.

You are expected to do the majority of the assignment outside the class meetings. Should you need assistance, or have questions about the assignment, please contact the instructor or the TA during their office hours.

You are encouraged to ask and answer questions on the course discussion board. (However, **do not share your answers** in the course discussion board.)

### Practice Exercises (no submission required)

1. Answer the multiple choice questions in text section 5.9 that have been covered in the lectures.

2 Complete the questions 2 through 5 in text section 5.9.

3 Look up the Unix command table in Appendix B.1 to find the meaning of these commands and then execute them: mkdir cse240prolog, cd cse240prolog, ls -l, chmod 700 cse240prolog, rm, rmdir, g++, gplc, pico, and vi.

4 Assuem that you have signed up the general server in early assignments. Log onto general.asu.edu and execute the commands you looked up.

5 Follow the Prolog tutorial in Text Appendix B.4 or the tutorials given in the course Web site, start GNU Prolog and try following simple programs (build-in functions). Don't forget the **period** at the end of the statement.

| ?- write(hello). % hello is a constant

| ?- write(Hello). % Hello is a variable. Its address will be displayed

| ?- write('hello world'). % a string is printed

| ?- read(Y), write('The variable entered is '), write(Y), nl.

/\* nl prints a newline. Type a period and an enter at the end of the input \*/

| ?- X is 2+2.

| ?- Y is 5\*8.

| ?- Y is 2\*\*10.

| ?- length([a, b, x, y, 2, 45, z], L).

| ?- append([a, b, c, d], [4, 6, 8], LL).

| ?- append(X,Y,[a,b,c]). Then, type ";" to obtain all possible answers.

| ?- X is [1 | [2 | [3 | []]]], write(X). Explain the output.

### Programming Assignment (50 points)

You are given the following Prolog (fact/rule) database. [30 points]

/\* Database for family. It consists of facts and rules. \*/

/\* Facts \*/

male(mark).

male(tom).

female(jen).

female(beth).

father\_of(mark, beth). /\* mark is the father of beth \*/

mother\_of(jen, tom). /\* jen is the mother of tom \*/

/\* Rules \*/

is\_male(X) :-

male(X);

father\_of(X, \_).

Enter the program using a text editor such as pico under Unix operation system and save the file as family.pl. You may enter the program on your own computer and upload the program into your cse240prolog directory in *general* server.

Compile the program using the Prolog command:

> gplc family.pl

Then, enter GNU Prolog programming environment by executing the Unix command gprolog.

Execute the program family by typing GNU Prolog command

|?- [family].

Ask questions by typing, e.g.:

|?- father\_of(mark, beth).

|?- mother\_of(beth, tom).

To exit from GNU Prolog, type your end-of-file character at the main Prolog prompt ^d (Ctrl-d).

| ?- ^d

You can find a complete set of GNU Prolog commands at http://www.gprolog.org/manual/gprolog.html.

Now, you can start to add your code into the program.

1.1 Add members to this family. Please pay close attention when adding these members, spelling counts and all letters should be lowercase. It may help to construct a family tree for this part. Please indent your additions so that it is easy to identify them.

Add the following males: eric, josh, and austin.

Add the following females: lisa, alice, and alex.

Add the following relationship facts:

* eric is the child of josh and lisa.
* alice is the child of austin and alex.
* jen and mark are both children of eric and alice. [5]

For all of the following questions, please label them. For example, if Question 1.0 asks you to define a rule called is\_male (X) that returns "yes" (or "true") if X is the father of a member of the family, then your code should look like:

/\* Question 1.0 \*/

is\_male(X) :-

male(X); % “;” is an “or” operation

father\_of(X, \_).

1.2 Define (add into the database) a rule called is\_female(X) that returns "yes" (or "true") if X is a female or the mother of a member of the family. [5]

Note: the system will return a "yes", if it finds a "true" answer and there exist no more true answers. The system will return "true ?" if it finds a "true" answer and there are still possibly further matches. In this case, if you type "enter", it will return "yes" and stop. If you type ";", it will continue to search for further answers.

1.3 Define a rule called grandmother\_of(X, Z) that returns “yes” (or “true”) if X is a grandmother of Z. Define another rule called grandfather\_of(X, Z) that returns “yes” (or “true”) if X is a grandfather of Z. [5]

1.4 Define a rule called sibling\_of(X, Y) that returns “yes” (or “true”) if X is a sibling of Y. [5]

1.5 Define a rule called parent\_of(X, Y) that returns “yes” (or “true”) if X is a parent of Y. [5]

1.6 Define a rule called descendent\_of(X, Y) that returns “yes” (or “true”) if X is a descendent of Y. Note: you will need to use recursion as well as the rule defined above. [5]

**2** Consider the following database that will be used to determine meals for halloween and thanksgiving with the ingredients needed for the given sides. [20 points]

|  |  |
| --- | --- |
| **halloween** | **thanksgiving** |
| *entre:* chocolate  *sides:*  candy   * sugar * foodcoloring   soda   * water * syrup | *entre:* turkey  *sides:*  bread   * egg * flour   pumpkinpie   * pumpkin * pie |

2.1 Create facts for the entres: entre(X, Y) where X is the occasion and Y is the entre.

Create facts for the side: side(X, Y) where X is the occasion and Y is the side.

Create facts for the ingredients : ingredient(X, Y) where X is the side and Y is the ingredient. [10]

2.2 Create a rule meal(X, Y) where X is the occasion and Y is the food that will be served. For example, meal(thanksgiving, X) should return turkey, bread, and pumpkinpie. [5]

2.3 Create a rule shoppinglist(X, Y) where X is the occasion and Y is the ingredient needed. For example, shoppinglist(thanksgiving, X) should return egg, flour, pumpkin, and pie. [5]

### Grading of Programming Assignment

The TA will grade your program following these steps:

(1) The TA will read your program and give points based on the points allocated to each component, the readability of your code (organization of the code and comments), logic, inclusion of the required functions, and correctness of the implementations of each function.

(2) Compile the code. If it does not compile, 20% of the points given in (1) will be deducted. For example, if you are given 20 points in step (1), your points will become 16 if the program fails to compile.

(3) If the code passes the compilation, the TA will execute and test the code. If, for any reason, the program gives an incorrect output or crashes for any input, 10% of the points given in (1) will be deducted.

### What to Submit?

You are required to submit your solutions in a compressed format (.zip). Make sure your compressed file is label correctly - lastname\_firstname12.zip.

The compressed file MUST contain the following files:

family.pl.

meals.pl.

No other files should be in the compressed folder.

If multiple submissions are made, the most recent submission will be graded. (Even if the assignment is submitted late.)

### Where to Submit?

All submissions must be electronically submitted to the respected homework link in the course web page where you downloaded the assignment.

### Late submission deduction policy

* No penalty for late submissions that are received within 24 hours after the deadline;
* 10% grade deduction for every day it is late after the grace period;
* No late submission after Tuesday at 11:59PM.