

## Programming Assignment #1: Prolog

Due at the beginning of the class on Monday, Oct. 20.

In order to complete this assignment, you must download and install SWI-Prolog (version 5.2) either on your own personal machine or in your home directory on one of the university machines. SWI-Prolog is free software, and can be downloaded from <http://www.swi-prolog.org>.

You will need to submit both electronic versions of your program files and hardcopies of both your source code and your interactions with the program.

- Your electronic versions of your program files must be submitted by using the course webpage on the Blackboard Learning System (see <https://ci.lehigh.edu>). From the CSE 262 page, select Assignments, and then click on “View/Complete”. You will then be able to attach each of your files for submissions. **Please make sure that you name the Prolog files exactly as requested.** We will be running scripts to process it and points may be taken off if the script fails to locate your code.
- Your hardcopies must include both source code and printouts of your interaction with the program. Please include a comment at the beginning of each Prolog file indicating your name and which problem it is the solution for. To show your interaction with Prolog, simply copy the relevant portions of the main Prolog window (your queries and Prolog’s answers) and paste it to a word processing file for printing.

1) **Simple knowledge base.** Represent the following facts in a Prolog knowledge base

- Ann likes every toy she plays with.
- A doll is a toy.
- A train is a toy.
- Ann plays with trains.
- John likes everything Ann likes.

Save the information in a file with either the name *<yourLastName>1.pl* or *<yourLastName>1.pro*. Then **consult** it to provide answers to the following:

- Who likes a doll?
- Who play with trains?
- Who does not play with trains?
- Is there anything John likes that Ann does not?
- Find all things that John likes.
- Find everyone who likes trains.

**Turn in your Prolog file and a printout of your interaction with the program.**

- 2) **List processing.** Write a Prolog program that defines the predicate `intersection(List1, List2, List3)`, where `List1`, `List2`, and `List3` are all lists of numbers, and `List3` is exactly the intersection of `List1` and `List2`. Save the information in a file with either the name `<yourLastName>2.pl` or `<yourLastName>2.pro`.

Use your program to find:

- a) the intersection of the list `[2,4,6,7]` and the list `[1, 2, 4]`
- b) the intersection of the list `[1,3,4]` and the list `[6, 9]`

**Hint:** This program will require recursion, and can be written with as few as three clauses.

**Turn in your Prolog file and a printout of your interaction with the program.**

- 3) **Solving word problems.** Develop a set of clauses describing the following word problem in such a way that Prolog can provide the solution:

Beth, Cindy, Felicia, Missy, and Sally live on different floors in a five-story building. Beth does not live on the 5<sup>th</sup> floor and Cindy does not live on the first. Felicia does not live on the top or the bottom floor, and she is not on a floor adjacent to Sally or Cindy. Missy lives on some floor above Cindy. Who lives on what floors?

The goal `building(X)` should return an instantiation for `X` that is the answer to the problem. This instantiation should be a list of five complex terms, each of the form `floor(Occupant, Number)`, where `Occupant` is the name of one of the women, and `Number` is the number of the floor they live on.

Save the information in a file with either the name `<yourLastName>3.pl` or `<yourLastName>3.pro`.

**Hints:**

1. Make sure you constrain the result so that there will be exactly one person on each floor.
2. Develop one or more rules that constrain the result based on the assertions made in the problem statement.
3. It may help to develop an extra predicate for determining whether two floors are adjacent.

**Turn in your Prolog file and a print out of your interaction with the program.** The interaction should just show your Prolog query, and the system's answer to the problem.