

## Installing GNU Prolog

We will be using GNU Prolog (`gprolog`) available at

<http://www.gprolog.org/>

There are binary distribution packages for Windows and Mac. Linux users can compile `gprolog` from the source.

The Mac distribution installs GNU Prolog in `/opt/local/bin`, so you will have to put that location in your path using the following steps.

1. `cd` to your home directory.
2. Edit your `.profile` file using the `vi` editor with the command

```
vi .profile
```

3. Add this line (see “vi Summary”)

```
PATH=/opt/local/bin:$PATH
```

4. If it does not already exist, add this line

```
export PATH
```

5. Quit the `vi` editor with the command

```
ZZ
```

## Using GNU Prolog

I recommend that you startup a text editor for your Prolog program and at the same time maintain a command line window with Prolog running. You can write your Prolog program and save it in your text editor, and then switch to the command line window to test it.

I highly recommend that you take this part of the class as an opportunity to learn how to use `vi`, the standard Unix text editor, for writing your Prolog programs. There are many beginning tutorials online. Here is a summary of the `vi` commands.

<http://www.cslab.pepperdine.edu/warford/cosc450/cosc-450-vi-Summary.pdf>

## Example GNU Prolog session

When the `gprolog` engine gives one solution and asks for further input with `?`, you have three options:

1. `;` to compute the next solution
2. `a` to compute all remaining solutions

3. <ret> to stop the execution

Here is a Prolog session from the first chapter.

```
Stans-Office-iMac:prolog warford$ gprolog
GNU Prolog 1.4.5 (64 bits)
Compiled Jul 14 2018, 16:25:15 with /usr/bin/clang
By Daniel Diaz
Copyright (C) 1999-2018 Daniel Diaz
| ?- consult( 'ch1.pl' ).
compiling /Users/warford/Documents/Classes/cs450/prolog/ch1.pl for byte code...
/Users/warford/Documents/Classes/cs450/prolog/ch1.pl compiled, 45 lines read -
2686 bytes written, 4 ms

(1 ms) yes
| ?- parent( bob, pat ).

yes
| ?- parent( liz, pat ).

no
| ?- parent( X, liz ).

X = tom ? a

no
| ?- parent( X, Y ).

X = pam
Y = bob ? ;

X = tom
Y = bob ? ;

X = tom
Y = liz ? ;

X = bob
Y = ann ? a

X = bob
Y = pat

X = pat
Y = jim

yes
```

You exit gprolog with the

```
halt.
```

command. The period is mandatory.

Code for the programs in the book are on the course web page. There is one requirement for gprolog programs not followed by our book. All clauses must be contiguous on the head of the clause. Thus, the program fragment from the book

```
female( pam).           % Pam is female
male( tom).             % Tom is male
male( bob).
female( liz).
female( ann).
female( pat).
male( jim).
```

must be written

```
female( pam).           % Pam is female
female( liz).
female( ann).
female( pat).
male( tom).             % Tom is male
male( bob).
male( jim).
```