ECE 479/579

- Please get an ECE id
- Go to https://account.ece.arizona.edu
- You will need this for turning in your projects, accessing stuff like Jess, receiving your grades etc.
- Brush up your Java skills
- Download and use Project#1+ code @ your own risk!

Assignment #1

- Jess (Java Expert System Shell) – It’s a set of Java classes which can be used in your own java code OR as a standalone expert system.
- Its very complex and we are going to be just getting an overview
- We will “not” be writing JAVA code for this assignment
- We will simply be making a “.clp” file which Jess takes as a command line argument.

Jess documentation

- When in doubt take a look at the Jess documentation
- Link to Jess homepage can be found in "class software" section of the ECE479/579 website.

Example

- All mammals are animals. All animals inhale oxygen (this is not biologically correct, but let's use it anyway). Sperm whales and elephants are mammals. Mammals are warm blooded. Elephants are grey, wrinkled and like peanuts. Sperm whales and sharks live in the ocean. Sharks are animals. All sharks and sperm whales are friends. Clyde is an elephant, John is a shark, and Henry is a sperm whale.

Example…… facts

```
% facts
(assert (elephant clyde))
(assert (shark john))
(assert (whale henry))
(assert (type henry sperm))
```

Example……. rules

```
% rules
(defrule rule-mammal-animal (mammal ?X) => (assert (animal
    ?X)))
(defrule rule-animal-inhales (animal ?X) => (assert (inhales ?X
    oxygen)))
(defrule rule-sperm-whale-animal (and (whale ?X) (type ?X sperm))
    => (assert (animal ?X)))
.
.
(defrule rule-shark-sperm-whale-friends (and (shark ?X) (whale ?Y)
    (type ?Y sperm)) => (assert (friends ?X ?Y)))
```

Example….. query

```
% define the query for finding whoever inhales oxygen
(defquery find-inhales "find inhales" (declare (variables ?Y))
    (inhales ?X ?Y))
% run the above query
(bind ?it (run-query find-inhales oxygen))
% print out the results by iterating
(while(?it hasNext)
(bind ?token (call ?it next))
(bind ?fact (call ?token fact 1))
(bind ?slot (fact-slot-value ?fact __data))
(bind ?datum (nth$ 1 ?slot))
(printout t ?datum crlf))
```

Artificial Intelligence….

```
Jess, the Java Expert System Shell
Copyright (C) 1998 E.J. Friedman Hill and the Sandia Corporation
Jess Version 6.0 12/7/2001
f-0 (MAIN::initial-fact)
f-1 (MAIN::elephant clyde)
f-2 (MAIN::shark john)
f-3 (MAIN::whale henry)
f-4 (MAIN::type henry sperm)
For a total of 5 facts.
==> f-5 (MAIN::friends john henry)
==> f-6 (MAIN::animal henry)
==> f-7 (MAIN::lives henry ocean)
==> f-8 (MAIN::inhales henry oxygen)
==> f-9 (MAIN::animal john)
==> f-10 (MAIN::inhales john oxygen)
==> f-11 (MAIN::lives john ocean)
==> f-12 (MAIN::favorite_food clyde peanuts)
==> f-13 (MAIN::mammal clyde)
==> f-14 (MAIN::animal clyde)
==> f-15 (MAIN::inhales clyde oxygen)
==> f-16 (MAIN::blood_temp clyde warm)
==> f-17 (MAIN::color clyde grey)
==> f-18 (MAIN::skin clyde wrinkled)
==> f-19 (MAIN::__query-trigger-find-inhales oxygen)
<== f-19 (MAIN::__query-trigger-find-inhales oxygen)
henry
john
clyde
```

The problem L

The family

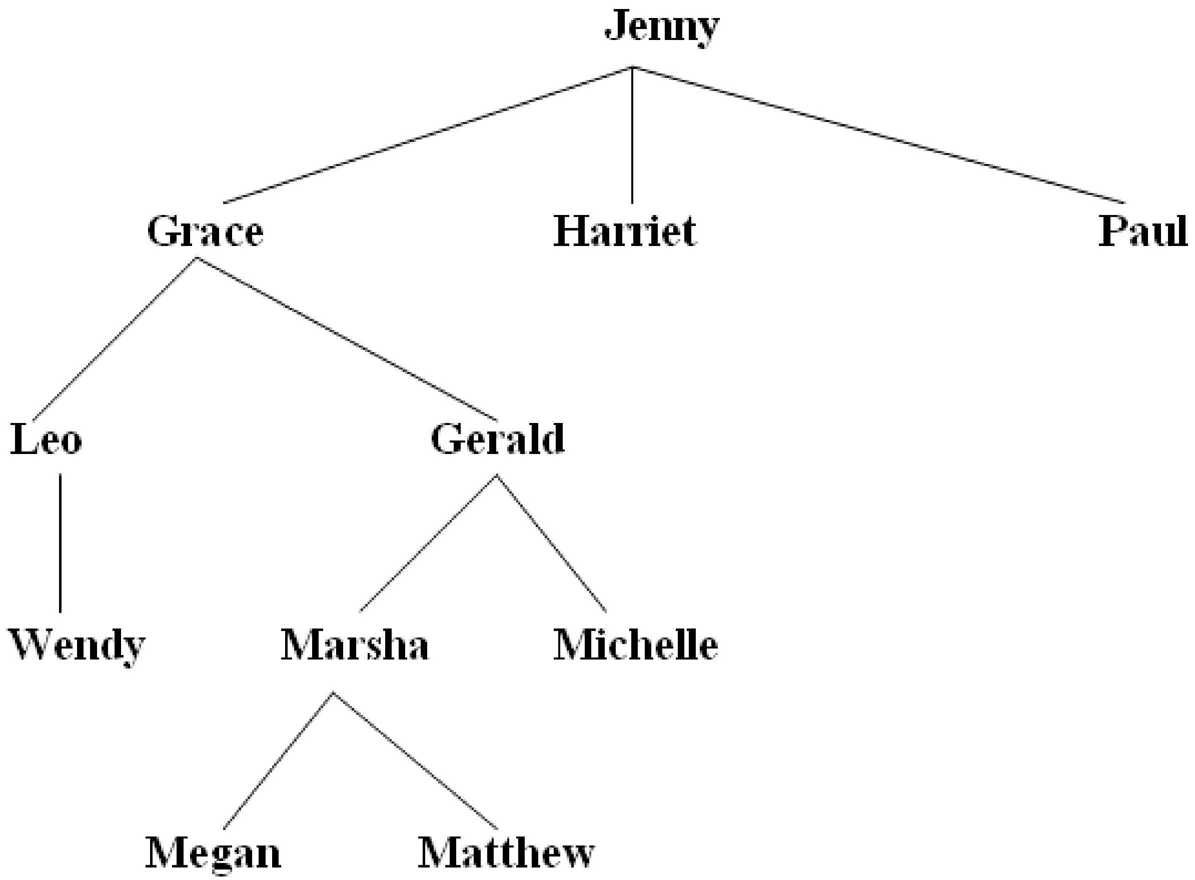
Note: if a person's name looks *emphasized*, then that person is female.

Jenny had three children: *Grace*, *Harriet*, and Paul. Neither *Harriet* nor Paul married. However, *Grace* married and had two children: Leo and Gerald.

Leo had one child: *Wendy* who never married

Gerald had twins: *Marsha* and *Michelle*. Although *Michelle* never married, *Marsha* married and had two children: *Megan* and Matthew.

The family tree….



Restrictions…

You are *only* allowed to use the primitives father(X,Y), mother(X,Y) and gender(X,Y) to encode the facts of the family tree. Also, you should not include unnecessary facts.

There is no one correct solution to the problem!

Questions to answer…

For the purposes of your assignment, you should submit one text file that contains your facts and rules of the above description. This file should be loadable by Jess and produce the intuitively correct results for any of the above predicates. However, in order to help you test your program you must try to answer the following:

- Who are cousins?
- Who is the great uncle of Kimberly?
- Who are aunts, uncles, grandfathers, grandmothers?
- Who are great-aunts, great-uncles, great-grandparents?

Each of the above should be made into a query, along with a appropriate iteration and print block as shown in the mammals example. These queries should be included in the family.clp file after all the facts and rules have been encoded.

Turn it in…..

```
turnin ece479 assn1 family.clp
```

Also print out a copy of your family.clp file and a sample run (with query results) and bring them to class to turn in on the due date of the assignment.

Need an extension ?