11 a. About Language (What/When/Who/Why)

```
% Facts about Prolog
language(prolog).
origin_year(1972).
creator('Alain Colmerauer').
purpose('Artificial Intelligence and computational linguistics').

% Rules to provide information about Prolog
about_language :-
   language(Lang),
   origin_year(Year),
   creator(Creator),
   purpose(Purpose),
   write('Language: '), write(Lang),nl,
   write('Origin Year: '), write(Year), nl,
   write('creator: '),write(Creator),nl,
   write('Purpose: '), write(Purpose), nl.

% Query to run the rule
% ?- about_language.
```

```
?- language(python).
false.

?- language(prolog).
true.

?- about_language.
Language: prolog
Origin Year: 1972
creator: Alain Colmerauer
Purpose: Artificial Intelligence and computational linguistics
true.
```

11 b. Atoms, Variables, Facts, and Rules in Prolog

```
% Facts about a family
parent(john, mary).      % john is a parent of mary
parent(mary, sam).       % mary is a parent of sam
parent(john, mike).      % john is a parent of mike
parent(mike, sara).      % mike is a parent of sara

% Rule to define grandparent relationship
grandparent(X, Y) :- parent(X, Z), parent(Z, Y).

% Atoms and Variables:
% Atom: john, mary, sam, mike, sara
% Variable: X, Y, Z

% Queries to run the rule
% ?- parent(john, mary).
% ?- grandparent(john, sam).
```

```
?- parent(john,mary).
true.

?- grandparent(john,sam).
true .

?- granndparent(john,sara).
Correct to: "grandparent(john,sara)"? yes
true.

?-
```

12. Ancestor program using Prolog

```
% Facts about the family tree
parent(john, mary).
parent(mary, sam).
parent(sam, kate).
parent(kate, alice).

% Rule to define ancestor relationship
ancestor(X, Y) :- parent(X, Y).
ancestor(X, Y) :- parent(X, Z), ancestor(Z, Y).

% Queries to test the rule
% ?- ancestor(john, alice).
% ?- ancestor(mary, kate).
% ?- ancestor(sam, alice).
```

```
File  Edit  Settings  Run  Debug  Help
% d:/materials/4th sem/ai/lab/part ii/question12 compiled 0.00 sec, -1 clauses
?- ancestor(john,mary).
true .

?- amcestor(john,alice).
Correct to: "ancestor(john,alice)"? yes
true .

?- ancestor(mary,kate).
true .

?-
```

13. Family relationship (family tree) program using prolog.

```
% Facts about the family tree
parent(john, mary).
parent(john, mike).
parent(susan, mary).
parent(susan, mike).
parent(mary, sam).
parent(tom, sam).
parent(mike, kate).
parent(linda, kate).
```

```
% Rules to define relationships
father(X, Y) :- parent(X, Y), male(X).
mother(X, Y) :- parent(X, Y), female(X).
sibling(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y.
brother(X, Y) :- sibling(X, Y), male(X).
sister(X, Y) :- sibling(X, Y), female(X).
grandparent(X, Y) :- parent(X, Z), parent(Z, Y).
grandfather(X, Y) :- grandparent(X, Y), male(X).
grandmother(X, Y) :- grandparent(X, Y), female(X).
ancestor(X, Y) :- parent(X, Y).
ancestor(X, Y) :- parent(X, Z), ancestor(Z, Y).
descendant(X, Y) :- ancestor(Y, X).
uncle(X, Y) :- brother(X, Z), parent(Z, Y).
aunt(X, Y) :- sister(X, Z), parent(Z, Y).

% Gender facts
male(john).
male(mike).
male(sam).
male(tom).
female(susan).
female(mary).
female(kate).
female(linda).

% Queries to test the rules
% ?- father(john, mary).
% ?- mother(susan, mike).
% ?- sibling(mary, mike).
% ?- brother(mike, mary).
% ?- sister(mary, mike).
% ?- grandfather(john, sam).
% ?- grandmother(susan, kate).
% ?- ancestor(john, sam).
% ?- descendant(sam, john).
% ?- uncle(mike, sam).
% ?- aunt(mary, kate).
```

```
?- father(john,mary).
true .

?- mother(susan,mike).
true.

?- sibling(mary,mike).
true .

?- sister(mary,mike).
true .

?- grandfather(john,sam).
true .

?- grandmother(susan,kate).
true.

?- ancestor(john,sam), descendant(sam,john), uncle(mike,sam), aunt(mary,kate).
true
```

14. Represent following facts in Semantic Net diagrammatically and also write a program in Prolog to represent the Semantic Net.

- Mat1 is a mat
- Cat1 is a cat
- Tom is a cat.
- Bird1 is a bird.
- Cat1 sat on Mat1.
- Tom caught bird1.
- Tom is owned by John.
- Tom is ginger in color.
- Cats like cream.
- The cat sat on the mat.
- A cat is a mammal.
- A bird is an animal.
- All mammals are animals.
- Mammals have fur.

Prolog Program to Represent the Semantic Net

% Facts

```
:- discontiguous is_a/2,has/2,likes/2.
is_a(mat1, mat).
is_a(cat1, cat).
is_a(tom, cat).
is_a(bird1, bird).
sat_on(cat1, mat1).
sat_on(the_cat, the_mat). % Assuming 'the_cat' is cat1 and 'the_mat' is mat1
caught(tom, bird1).
owned_by(tom, john).
color(tom, ginger).
likes(cat,cream).

is_a(cat, mammal).
is_a(bird, animal).
is_a(mammal, animal).
has(mammal, fur).

% Rules to infer relationships
likes(X, cream) :- is_a(X,cat).
is_a(X,mammal) :- is_a(X,cat).
is_a(X, animal) :- is_a(X, mammal).
is_a(X, animal) :- is_a(X, bird).
has(X,fur) :- is_a(X,mammal).


% Queries
% Check if Tom is an animal
%?- is_a(tom, animal).

% Find out who sat on the mat
%?- sat_on(X, mat1).


% Determine the color of Tom
%?- color(tom, Color).

% Check if Cat1 likes cream
%?- likes(cat1, cream).

% Find all animals in the Semantic Net
%?- is_a(X, animal).

% Check who caught Bird1
%?- caught(X, bird1).

% Find out who owns Tom
```

%?- owned_by(tom, Owner).

% Check if Mammals have fur
%?- has(mammal, fur).

% Find the relationships of Tom
%?- is_a(tom, X).

SWI-Prolog -- d:/Materials/4th sem/AI/lab/part ii/question11answer.pl

File Edit Settings Run Debug Help

```
% d:/materials/4th sem/ai/lab/part ii/question14 compiled 0.00 sec, 0 clauses
?-
|    is_a(tom,animal).
true .

?- likes(cat1,cream).
true .

?- color(tom,Color).
Color = ginger.

?- is_a(X,animal).
X = bird .

?- is_a(X,animal).
X = bird ;
X = mammal ;
X = cat ;
X = cat1 ;
X = tom ;
X = bird1.

?- is_a(tom,X).
X = cat ;
X = mammal ;
X = animal ;
false.

?-
```