Errol Quinones CSC344 – Programming Languages

Prolog Assignment #1: Various Computations

ABSTRACT:

This first Prolog assignment serves as an introduction to Prolog programming through interaction and practice with knowledge bases (KB). These tasks contain KBs pertaining to colors, foods, and shapes.

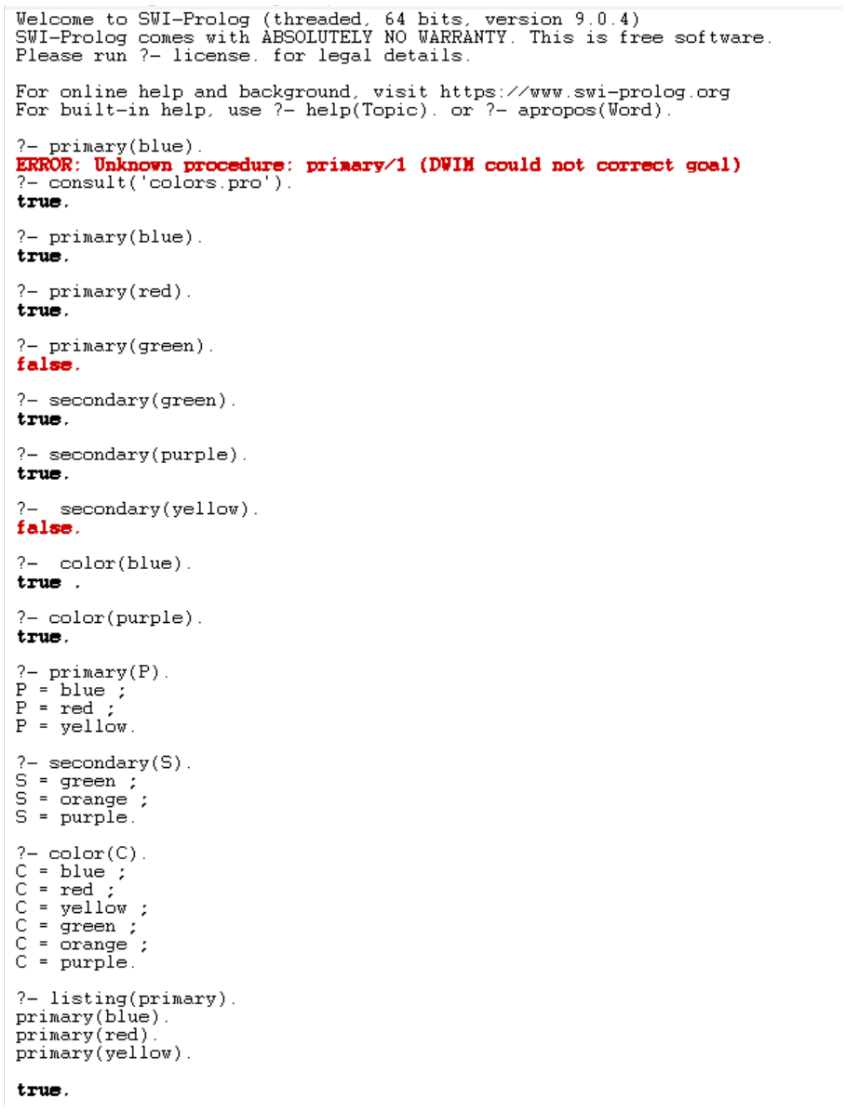
Task 1: Colors KB

Colors KB Code

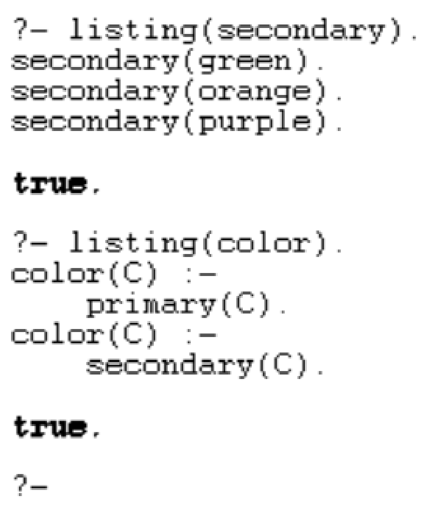
```
% ---------------------------------------------------------------------
% File: colors.pro
% Line: Six color facts, structured into primaries and secondaries
% ---------------------------------------------------------------------
% primary(P) :: P is a primary color
primary(blue).
primary(red).
primary(yellow).
% ---------------------------------------------------------------------
% secondary(S) :: S is a secondary color
secondary(green).
secondary(orange).
secondary(purple).
```

```
% ------------------------------------------------------------------
% color(C) :: C is a color
color(C) :- primary(C).
color(C) :- secondary(C).
```

Colors KB Interaction



```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- primary(blue).
ERROR: Unknown procedure: primary/1 (DWIM could not correct goal)
?- consult('colors.pro').
true.

?- primary(blue).
true.

?- primary(red).
true.

?- primary(green).
false.

?- secondary(green).
true.

?- secondary(purple).
true.

?-  secondary(yellow).
false.

?-  color(blue).
true .

?- color(purple).
true.

?- primary(P).
P = blue ;
P = red ;
P = yellow.

?- secondary(S).
S = green ;
S = orange ;
S = purple.

?- color(C).
C = blue ;
C = red ;
C = yellow ;
C = green ;
C = orange ;
C = purple.

?- listing(primary).
primary(blue).
primary(red).
primary(yellow).

true.
```



```
?- listing(secondary).
secondary(green).
secondary(orange).
secondary(purple).

true.

?- listing(color).
color(C) :-
    primary(C).
color(C) :-
    secondary(C).

true.

?-
```

Task 2: Food KB

Food KB Code

% fruit(F) :: F is a fruit

fruit(grapefruit).

fruit(avocado).

fruit(date).

% ---

% vegetable(V) :: V is a vegetable

vegetable(asperagus).

vegetable(broccoli).

vegetable(carrot).

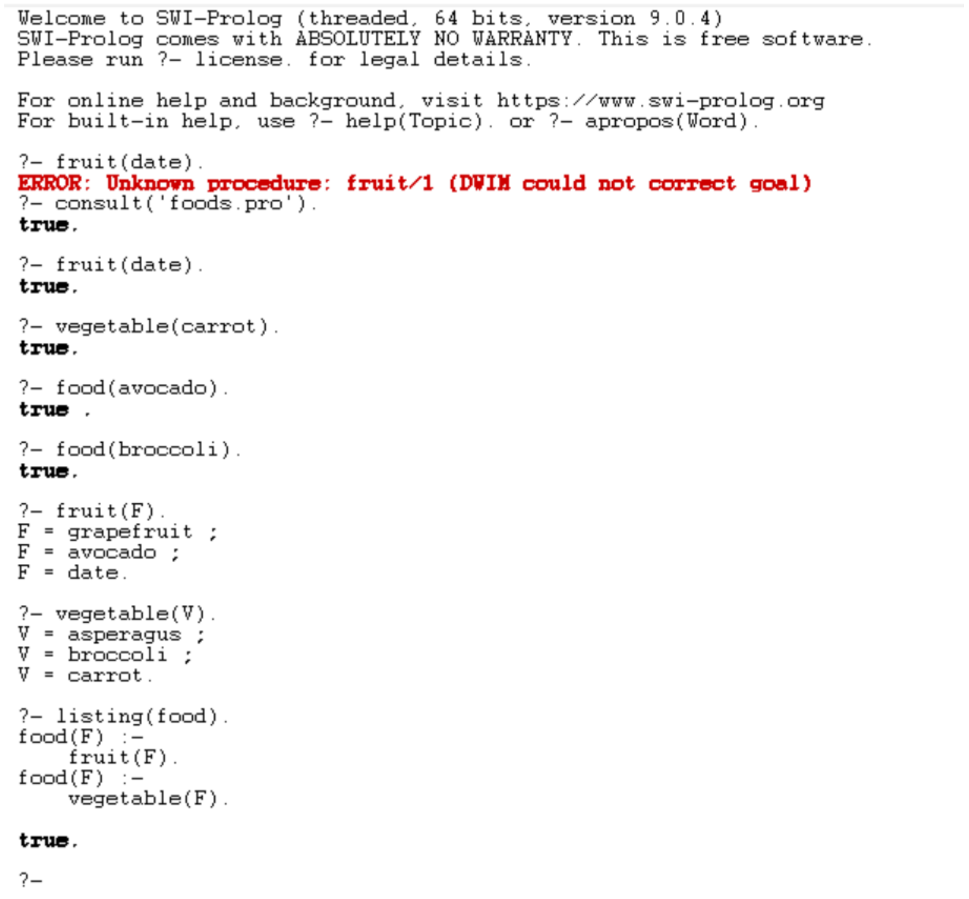
% ---

% food(F) :: F is a food

food(F) :- fruit(F).

food(F) :- vegetable(F).

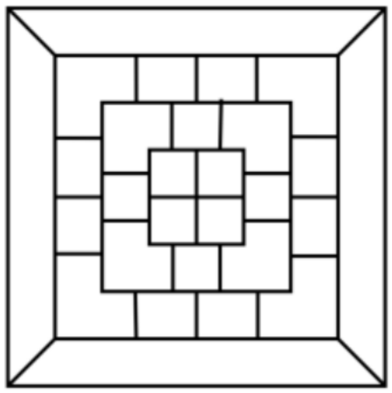
Food KB Interaction

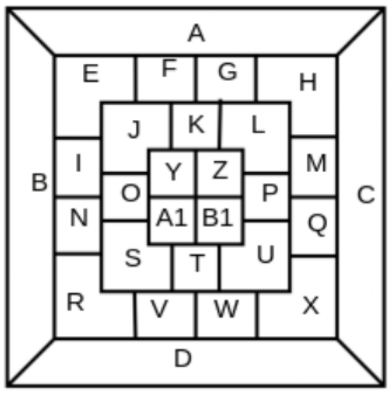


```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- fruit(date).
ERROR: Unknown procedure: fruit/1 (DWIM could not correct goal)
?- consult('foods.pro').
true.

?- fruit(date).
true.

?- vegetable(carrot).
true.

?- food(avocado).
true .

?- food(broccoli).
true.

?- fruit(F).
F = grapefruit ;
F = avocado ;
F = date.

?- vegetable(V).
V = asparagus ;
V = broccoli ;
V = carrot.

?- listing(food).
food(F) :-
    fruit(F).
food(F) :-
    vegetable(F).

true.

?-
```

Task 3: Map Coloring

The Given Map ***The Labeled Map***



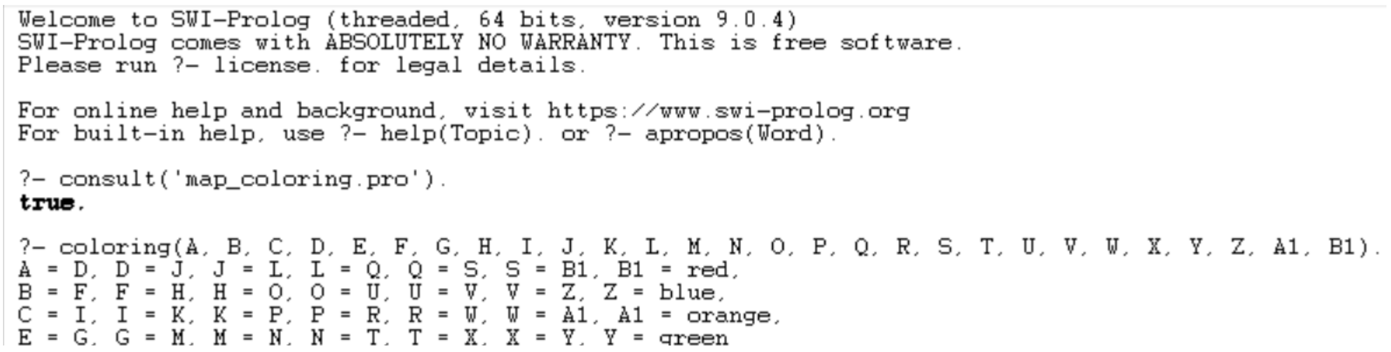


Code For Coloring the Map

```
% different(X,Y) :: X is not equal to Y

different(red,blue).

different(red,green).

different(red,orange).

different(green,blue).

different(green,orange).

different(green,red).

different(blue,green).

different(blue,orange).

different(blue,red).

different(orange,blue).

different(orange,green).

different(orange,red).

coloring(A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A1,
B1) :-

different(A, B),

different(A, E),
different(A, F),
different(A, G),
different(A, H),
different(A, C),
different(B, E),
different(B, I),
different(B, N),
different(B, R),
different(B, D),
different(D, V),
different(D, W),
different(D, X),
different(D, C),
different(D, R),
different(C, X),
different(C, Q),
different(C, M),
different(C, H),
different(E, F),
different(E, J),
different(E, I),
different(F, J),
different(F, K),
different(F, G),
different(G, K),
different(G, L),
different(G, H),
different(H, L),
different(H, M),
different(M, L),
different(M, P),
different(M, Q),
different(Q, P),
different(Q, U),
different(Q, X),
different(X, U),
different(X, W),
different(W, U),
different(W, T),
different(W, V),
different(V, T),
different(V, S),
different(V, R),
different(R, S),
different(R, N),
different(N, S),
different(N, O),
different(N, I),
different(I, O),
different(I, J),
different(J, O),
different(J, Y),
different(J, K),
different(K, Y),
different(K, Z),
different(K, L),
different(L, Z),
different(L, P),
different(P, Z),
different(P, B1),
different(P, U),
different(U, B1),
different(U, T),
different(T, B1),
different(T, A1),
different(T, S),
different(S, A1),
different(S, O),
different(O, A1),
different(O, Y),
different(Y, Z),
different(Y, B1),
different(Y, A1),
different(Z, A1),
different(Z, B1),
```

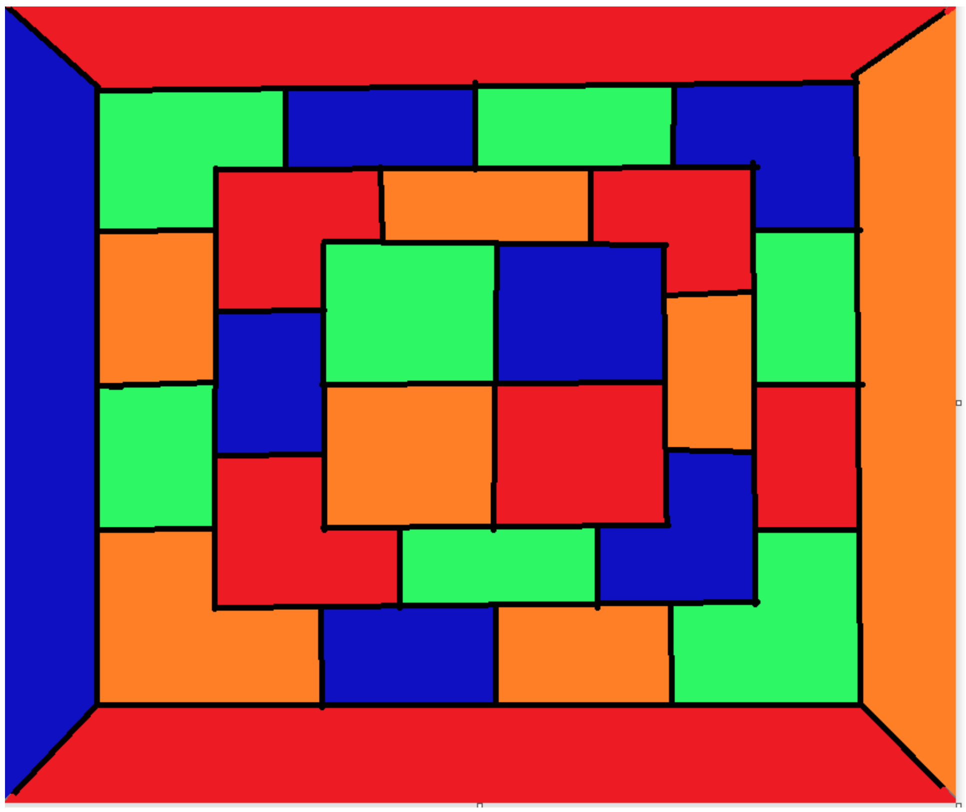
different(A1, B1).

Map Coloring Interaction



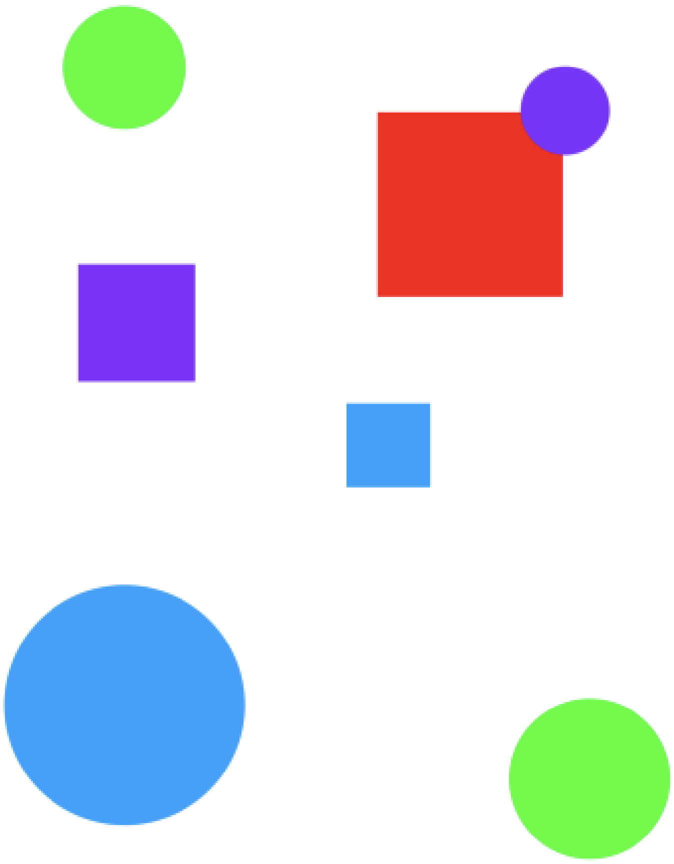
```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('map_coloring.pro').
true.

?- coloring(A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A1, B1).
A = D, D = J, J = L, L = Q, Q = S, S = B1, B1 = red,
B = F, F = H, H = O, O = U, U = V, V = Z, Z = blue,
C = I, I = K, K = P, P = R, R = W, W = A1, A1 = orange,
E = G, G = M, M = N, N = T, T = X, X = Y, Y = green
```

The Colored Map



Task 4: Floating Shapes World KB

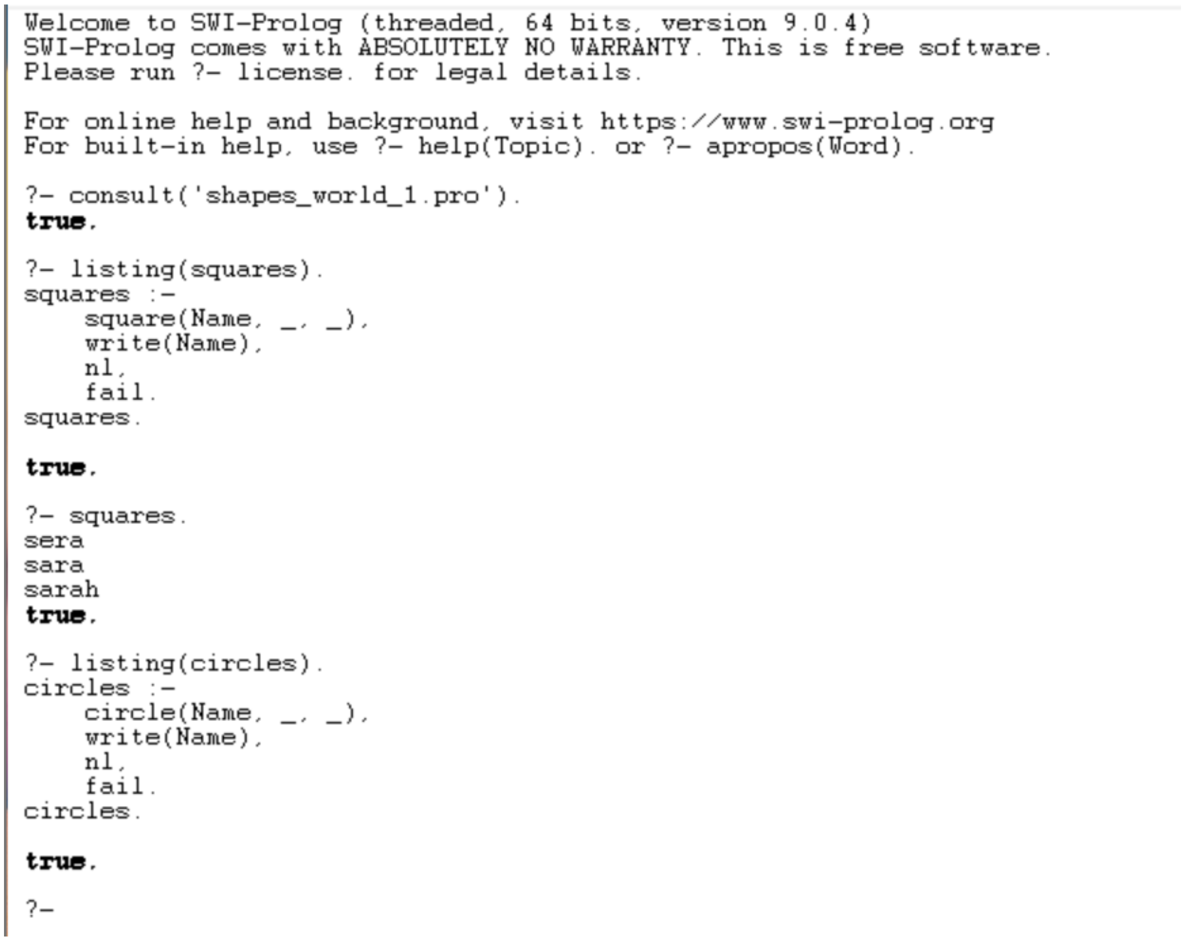
Floating Shapes World Image



Floating Shapes World KB Code

```
% ----------------------------------------------------------------------
% ----------------------------------------------------------------------
% --- File: shapes_world_1.pro
% --- Line: Loosely represented 2-D shapes world (simple take on SHRDLU)
% ----------------------------------------------------------------------
% ----------------------------------------------------------------------
% --- Facts ...
% ----------------------------------------------------------------------
% --------------------------------------------------------------------
% --- square(N,side(L),color(C)) :: N is the name of a square with side L
% --- and color C
square(sera,side(7),color(purple)).
square(sara,side(5),color(blue)).
square(sarah,side(11),color(red)).
% --------------------------------------------------------------------
% --- circle(N,radius(R),color(C)) :: N is the name of a circle with
% --- radius R and color C
circle(carla,radius(4),color(green)).
circle(cora,radius(7),color(blue)).
circle(connie,radius(3),color(purple)).
circle(claire,radius(5),color(green)).
% --------------------------------------------------------------------
% Rules ...
% --------------------------------------------------------------------
% --------------------------------------------------------------------
% --- circles :: list the names of all of the circles
circles :- circle(Name,_,_), write(Name),nl,fail.
circles.
% --------------------------------------------------------------------
% --- squares :: list the names of all of the squares
squares :- square(Name,_,_), write(Name),nl,fail.
squares.
% --------------------------------------------------------------------
% --- squares :: list the names of all of the shapes
shapes :- circles,squares.
% -------------------------------------------------------------------
% --- blue(Name) :: Name is a blue shape
blue(Name) :- square(Name,_,color(blue)).
blue(Name) :- circle(Name,_,color(blue)).
% -------------------------------------------------------------------
% --- large(Name) :: Name is a large shape
large(Name) :- area(Name,A), A >= 100.
% -------------------------------------------------------------------
% --- small(Name) :: Name is a small shape
small(Name) :- area(Name,A), A < 100.
% -------------------------------------------------------------------
% --- area(Name,A) :: A is the area of the shape with name Name
area(Name,A) :- circle(Name,radius(R),_), A is 3.14 * R * R.
area(Name,A) :- square(Name,side(S),_), A is S * S.
```

Floating Shapes World KB Interaction



```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- consult('shapes_world_1.pro').
true.

?- listing(squares).
squares :-
    square(Name, _, _),
    write(Name),
    nl,
    fail.
squares.

true.

?- squares.
sera
sara
sarah
true.

?- listing(circles).
circles :-
    circle(Name, _, _),
    write(Name),
    nl,
    fail.
circles.

true.

?-
```