```
% Frames for animals categories.
frame(mammal, [hair(yes), warmBlooded(yes), reproduction(birth), wings(no)]).
frame(bird, [hair(no), warmBlooded(yes), reproduction(egg), wings(yes)]).
frame(reptile, [hair(no), warmBlooded(no), reproduction(egg), wings(no)]).

% Frames for some animals.
frame(dolphin, [legs(0), habitat(water), movement(swim), flies(no)]).
frame(eagle, [legs(2), habitat(land), movement(flies) , flies(yes)]).
frame(turtle, [legs(4), habitat(land), habitat(water), movement(crawl), flies(no)]).

% inheritances
inherits_from(dolphin, mammal).
inherits_from(eagle, bird).
inherits_from(turtle, reptile).

% find a value from the animal frame or the animal's category frame(inheritance).
has(Animal, Value):- frame(Animal, Values), member(Value,Values).
has(Animal, Value):- inherits_from(Animal, X), frame(X, _), has(X, Value).

% member function.
member(Value, [Value|_]).
member(Value, [X|Y]):- member(Value, Y).

% classification rules.
mammal(X):- has(X, hair(yes)), has(X, warmBlooded(yes)).
bird(X):- has(X,wings(yes)), has(X, reproduction(egg)).
reptile(X):- has(X, movement(crawl)), has(X, warmBlooded(no)).

% animal type.
animal_type(X,Y):- (Y = mammal , mammal(X)); (Y = bird, bird(X)); (Y = reptile, reptile(X)).
```

ITCS440 ASSIGNMENT 1 | MOHAMED JAAFAR ABDULLA KHALAF | 202107999

SWI-Prolog (AMD64, Multi-threaded, version 9.3.13)

File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.3.13)
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).

?-
Warning: c:/users/mohdj/desktop/uob/prolog/example100.pl:22:
Warning:    Singleton variables: [X]
% c:/Users/mohdj/Desktop/UOB/PROLOG/example100.pl compiled 0.00 sec, 17 clauses
?- mammal(dolphin).
true .

?- bird(eagle).
true .

?- reptile(turtle).
true .

?- reptile(eagle).
false.

?- has(dolphin,habitat(X)).
X = water .

?- has(turtle,movement(X)).
X = crawl .

?- has(eagle,flies(X)).
X = yes .

?- has(eagle,legs(X)).
X = 2 .

?- animal_type(turtle,X).
X = reptile .

?- has(dolphin,warmBlooded(X)).
X = yes .

?- has(dolphin,movement(X)).
X = swim .
```