

Logic Programming Lab

Program

```
1 food(burger).
2 food(sandwich).
3 food(pizza).
4 lunch(sandwich).
5 dinner(pizza).
6 meal(X):-food(X).
7
8
```

food(pizza)

true

meal(X),lunch(X).

X = sandwich

dinner(sandwich).

false

```
8 studies(charlie, csc135).
9 studies(olivia, csc135).
10 studies(jack, csc131).
11 studies(arthur, csc134).
12 teaches(kirke, csc135).
13 teaches(collins, csc131).
14 teaches(collins, csc171).
15 teaches(juniper, csc134).
16 professor(X,Y):-teaches(X,C),studies(Y,C).
```

studies(charlie,C).

C = csc135

professor(kirke,Y).

Y = charlie

Y = olivia

```
12 teaches(kirke, csc135).
13 teaches(collins, csc131).
14 teaches(collins, csc171).
15 teaches(juniper, csc134).
16 professor(X,Y):-teaches(X,C),studies(Y,C).
17
18 owns(jack, car(bmw)).
19 owns(john, car(chevy)).
20 owns(olivia, car(civic)).
21 owns(jane, car(chevy)).
22 sedan(X):- X = car(bmw) ; X = car(civic).
23 truck(X):- X = car(chevy).
24
25 cat(fubby).
26 blackspot(fubby).
27 dog(figaro).
28 whitespot(figaro).
29 ownsPet(mary,X):-cat(X),blackspot(X).
30 loves(Y,X):-ownsPet(Y,X).
31
32 studies(charlie, csc135).
33 studies(olivia, csc135).
34 studies(jack, csc131).
35 studies(arthur, csc134).
36 teaches(kirke, csc135).
37 teaches(collins, csc131).
38 teaches(collins, csc171).
39 teaches(juniper, csc134).
40 professor(X,Y):-teaches(X,C),studies(Y,C).
41
42 owns(jack, car(bmw)).
43 owns(john, car(chevy)).
44 owns(olivia, car(civic)).
45 owns(jane, car(chevy)).
46 sedan(X):- X = car(bmw) ; X = car(civic).
47 truck(X):- X = car(chevy).
48
49 cat(fubby).
50 blackspot(fubby).
51 dog(figaro).
52 whitespot(figaro).
53 ownsPet(mary,X):-cat(X),blackspot(X).
54 loves(X,Y):-ownsPet(X,Y).
```

The screenshot shows a Prolog interpreter window with the following content:

```
owns(john,X).
X = car(chevy)

owns(john,_).
true

owns(Who,car(chevy)).
Who = john
Who = jane

owns(jane,X),sedan(X).
false

owns(jane,X),truck(X).
X = car(chevy)

?- owns(jane,X),truck(X).
```

Below this, there is another section of the interpreter window showing the execution of the `listing` predicate:

```
listing(cat).
cat(fubby).

true

listing(ownsPet).
ownsPet(mary, X) :-
    cat(X),
    blackspot(X).

true

loves(Who,What).
What = fubby,
Who = mary

ownsPet(mary,_).
true

?- ownsPet(mary,_).
```