Lab 12

Artificial Intelligence

BY

Muhammad Naqeeb

FA20-BCS-012

TO

Sir Mr. Qazi Zia

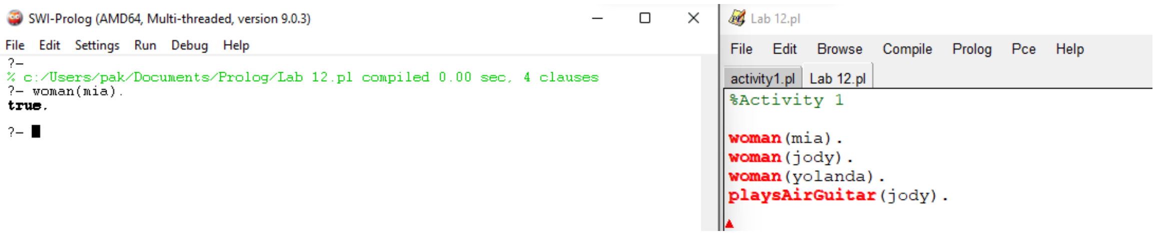
DEPARTMENT OF COMPUTER SCIENCE

COMSATS INSTITUTE OF INFORMATION AND TECHNOLOGY ATTOCK CAMPUS.

Lab 12
Introduction to Prolog

Activity 1:

How to add facts in a Knowledge Base?


SWI-Prolog (AMD64, Multi-threaded, version 9.0.3)
File Edit Settings Run Debug Help
Lab 12.pl
File Edit Browse Compile Prolog Pce Help
activity1.pl Lab 12.pl

```
?-
% c:/Users/pak/Documents/Prolog/Lab 12.pl compiled 0.00 sec, 4 clauses
?- woman(mia).
true.

?- 
```

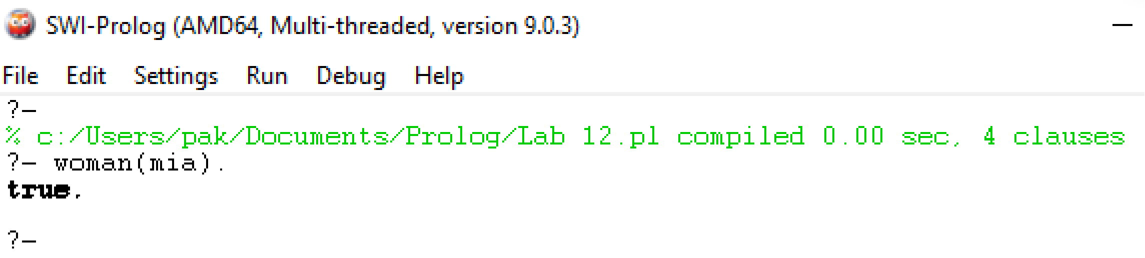
```
%Activity 1

woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
```

```
woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
```

Activity 2:

How to run a query within a Knowledge Base?

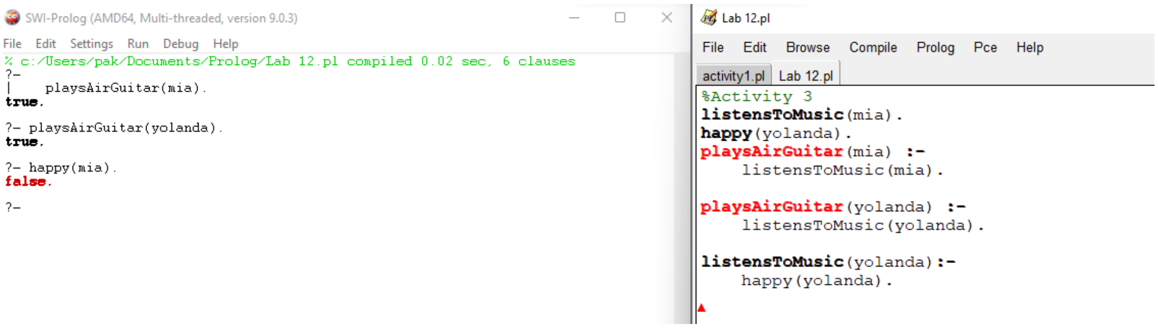

SWI-Prolog (AMD64, Multi-threaded, version 9.0.3)
File Edit Settings Run Debug Help

```
?-
% c:/Users/pak/Documents/Prolog/Lab 12.pl compiled 0.00 sec, 4 clauses
?- woman(mia).
true.

?-
```

Activity 3:

How to add rules alongside facts in a Knowledge Base?


SWI-Prolog (AMD64, Multi-threaded, version 9.0.3)
File Edit Settings Run Debug Help
Lab 12.pl
File Edit Browse Compile Prolog Pce Help
activity1.pl
Lab 12.pl

```
% c:/Users/pak/Documents/Prolog/Lab 12.pl compiled 0.02 sec, 6 clauses
?-
|    playsAirGuitar(mia).
true.

?- playsAirGuitar(yolanda).
true.

?- happy(mia).
false.

?-
```

```
%Activity 3
listensToMusic(mia).
happy(yolanda).
playsAirGuitar(mia)  :-
    listensToMusic(mia).

playsAirGuitar(yolanda)  :-
    listensToMusic(yolanda).

listensToMusic(yolanda):-
    happy(yolanda).
```

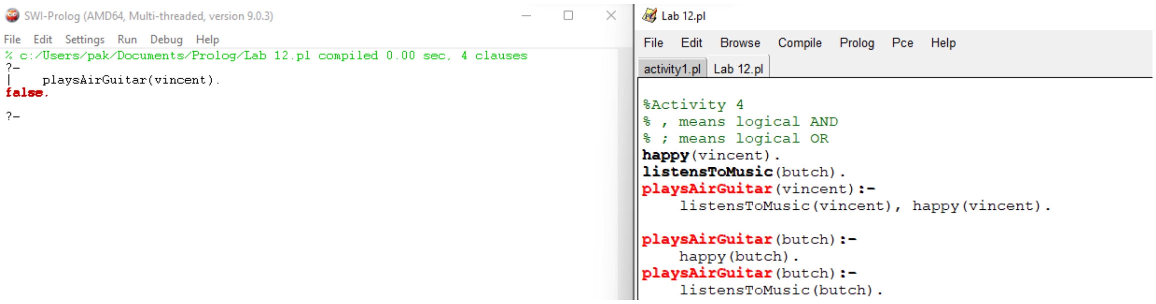
```
listensToMusic(mia).
happy(yolanda).
playsAirGuitar(mia)  :-
    listensToMusic(mia).

playsAirGuitar(yolanda)  :-
    listensToMusic(yolanda).

listensToMusic(yolanda):-
    happy(yolanda).
```

Activity 4:

Use conjunction-based rules in a Knowledge Base?


SWI-Prolog (AMD64, Multi-threaded, version 9.0.3)
File Edit Settings Run Debug Help
Lab 12.pl
File Edit Browse Compile Prolog Pce Help
activity1.pl
Lab 12.pl

```
% c:/Users/pak/Documents/Prolog/Lab 12.pl compiled 0.00 sec, 4 clauses
?-
|    playsAirGuitar(vincent).
false.

?-
```

```
%Activity 4
% , means logical AND
% ; means logical OR
happy(vincent).
listensToMusic(butch).
playsAirGuitar(vincent):-
    listensToMusic(vincent), happy(vincent).

playsAirGuitar(butch):-
    happy(butch).
playsAirGuitar(butch):-
    listensToMusic(butch).
```

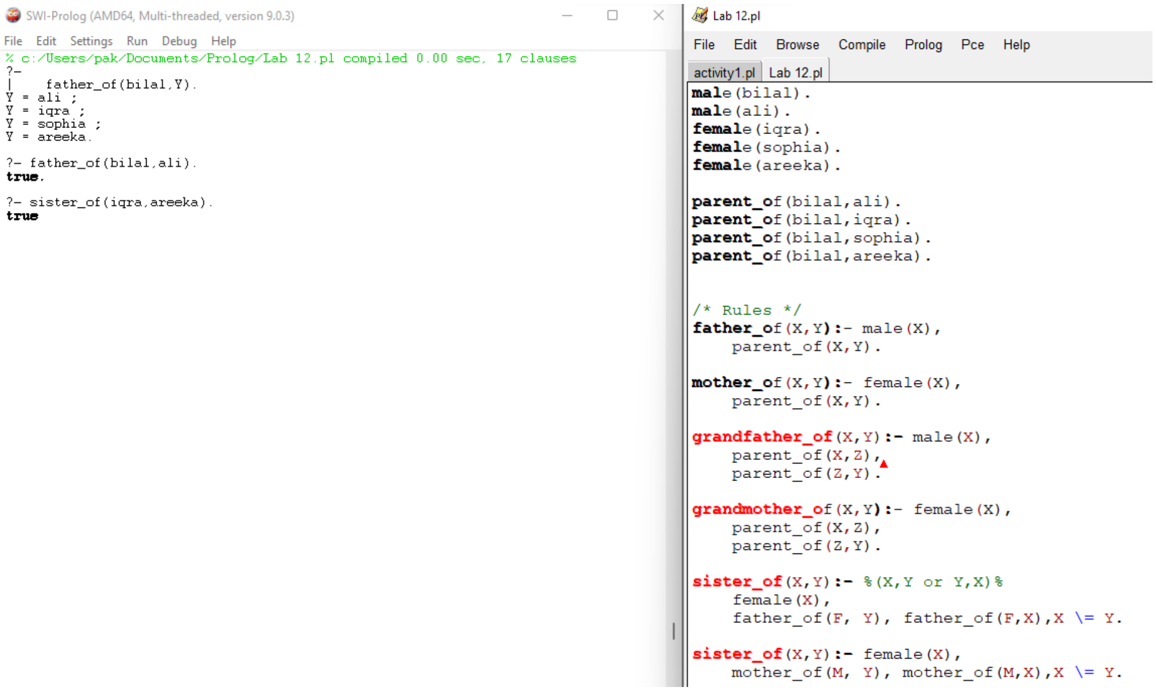
```
%Activity 4
% , means logical AND
% ; means logical OR
happy(vincent).
listensToMusic(butch).
playsAirGuitar(vincent):-
    listensToMusic(vincent), happy(vincent).

playsAirGuitar(butch):-
```

```
    happy(butch).
playsAirGuitar(butch):-
    listensToMusic(butch).
```

Lab Task 1

Create a knowledge base which defines your family tree and make a query that uses application of modus ponens to derive a fact which is not explicitly elaborated in the knowledge base.


SWI-Prolog (AMD64, Multi-threaded, version 9.0.3)
File Edit Settings Run Debug Help

```
% c:/Users/pak/Documents/Prolog/Lab 12.pl compiled 0.00 sec, 17 clauses
?-
|    father_of(bilal,Y).
Y = ali ;
Y = iqra ;
Y = sophia ;
Y = areeka.

?- father_of(bilal,ali).
true.

?- sister_of(iqra,areeka).
true
```

Lab 12.pl
File Edit Browse Compile Prolog Pce Help
activity1.pl
Lab 12.pl

```
male(bilal).
male(ali).
female(iqra).
female(sophia).
female(areeka).

parent_of(bilal,ali).
parent_of(bilal,iqra).
parent_of(bilal,sophia).
parent_of(bilal,areeka).


/* Rules */
father_of(X,Y):- male(X),
    parent_of(X,Y).

mother_of(X,Y):- female(X),
    parent_of(X,Y).

grandfather_of(X,Y):- male(X),
    parent_of(X,Z),
    parent_of(Z,Y).

grandmother_of(X,Y):- female(X),
    parent_of(X,Z),
    parent_of(Z,Y).

sister_of(X,Y):- %(X,Y or Y,X)%
    female(X),
    father_of(F, Y), father_of(F,X),X \= Y.

sister_of(X,Y):- female(X),
    mother_of(M, Y), mother_of(M,X),X \= Y.
```

```
/* Facts */
male(bilal).
male(ali).
female(iqra).
female(sophia).
female(areeka).

parent_of(bilal,ali).
parent_of(bilal,iqra).
parent_of(bilal,sophia).
parent_of(bilal,areeka).


/* Rules */
father_of(X,Y):- male(X),
    parent_of(X,Y).

mother_of(X,Y):- female(X),
    parent_of(X,Y).

grandfather_of(X,Y):- male(X),
    parent_of(X,Z),
    parent_of(Z,Y).

grandmother_of(X,Y):- female(X),
    parent_of(X,Z),
    parent_of(Z,Y).

sister_of(X,Y):- %(X,Y or Y,X)%
    female(X),
    father_of(F, Y), father_of(F,X),X \= Y.

sister_of(X,Y):- female(X),
    mother_of(M, Y), mother_of(M,X),X \= Y.


brother_of(X,Y):- %(X,Y or Y,X)%
    male(X),
    father_of(F, Y), father_of(F,X),X \= Y.

brother_of(X,Y):- male(X),
    mother_of(M, Y), mother_of(M,X),X \= Y.
```