Principles of Programming Language (CS302)

Assignment - 7

**U19CS012**

1.) Write the predicates using the following relations:

1. parent/2 -

**parent(X, Y)** [parent predicate with 2 arguments]

Read as: - *X is the Parent of Y.*

1. male/1

**male(X)** [male predicate with 1 argument]

Read as: - *X is Male.*

1. female/1

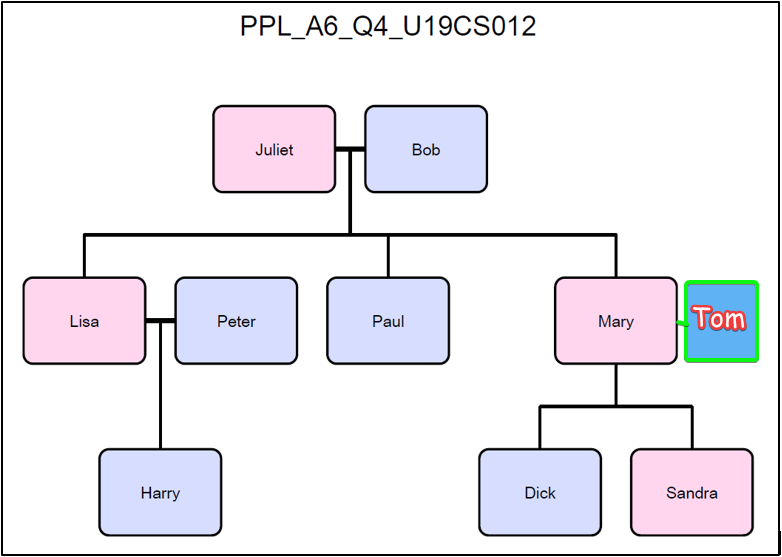
**female(X)** [female predicate with 1 argument]

Read as: - *X is Female.*

1. husband/2

**husband(X,Y)** [husband predicate with 2 arguments]

Read as: - *X is the Husband of Y.*



2.) Write the rules using above relations for the following:

**PROLOG Code**

*/\* Facts \*/*

female(mary)*.*

female(sandra)*.*

female(juliet)*.*

female(lisa)*.*

male(peter)*.*

male(paul)*.*

male(dick)*.*

male(bob)*.*

male(harry)*.*

*% Added "tom", Husband of Mary*

male(tom)*.*

parent(bob, lisa)*.*

parent(bob, paul)*.*

parent(bob, mary)*.*

parent(juliet, lisa)*.*

parent(juliet, paul)*.*

parent(juliet, mary)*.*

parent(peter, harry)*.*

parent(lisa, harry)*.*

parent(mary, dick)*.*

parent(mary, sandra)*.*

husband(bob,juliet)*.*

husband(peter,lisa)*.*

husband(tom,mary)*.*

*/\* Rules \*/*

*% Child*

child(X, Y)*:-*

    parent(Y, X)*.*

*% Mother*

mother(X, Y)*:-*

    female(X),

    parent(X, Y)*.*

*% Father*

father(X, Y)*:-*

    male(X),

    parent(X, Y)*.*

*% Wife*

wife(X, Y)*:-*

    husband(Y, X)*.*

*% Son*

son(X, Y)*:-*

    male(X),

    parent(Y, X)*.*

*% Daughter*

daughter(X, Y)*:-*

    female(X),

    parent(Y, X)*.*

*% Brother*

brother(X,Y) *:-*

    male(X),

    parent(P,X),

    parent(P,Y),

    X \= Y*.*

print\_brothers *:-* brother(X, Y), format('~a is\_brother\_of ~a', [X, Y])*.*

*% Sister*

sister(X,Y) *:-*

    female(X),

    parent(P,X),

    parent(P,Y),

    X \= Y*.*

print\_sisters *:-* sister(X, Y), format('~a is\_sister\_of ~a', [X, Y])*.*

*% Uncle - X is an uncle of Y if X is the brother of some person, Z, and Z is a parent of Y*

uncle(X, Y) *:-*

    brother(X, Z),

    parent(Z, Y)*.*

print\_uncles *:-* uncle(X, Y), format('~a is\_uncle\_of ~a', [X, Y])*.*

*% Aunt - X is an aunt of Y if X is the sister of some person, P, and P is a parent of Y*

aunt(X, Y) *:-*

    sister(X, P),

    parent(P, Y)*.*

aunt(X, Y) *:-*

    parent(P, Y),

    brother(Uncle, P),

    wife(X, Uncle)*.*

print\_aunts *:-* aunt(X, Y), format('~a is\_aunt\_of ~a', [X, Y])*.*

*% Nephew*

nephew(X, Y) *:-* uncle(Y, X), male(X)*.*

nephew(X, Y) *:-* aunt(Y, X), male(X)*.*

print\_nephews *:-* nephew(X, Y), format('~a is\_nephew\_of ~a', [X, Y])*.*

*% Niece*

niece(X, Y) *:-* uncle(Y, X), female(X)*.*

niece(X, Y) *:-* aunt(Y, X), female(X)*.*

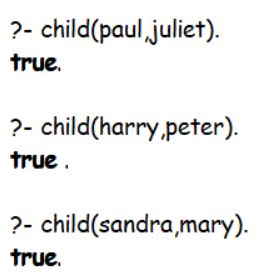
print\_nieces *:-* niece(X, Y), format('~a is\_niece\_of ~a', [X, Y])*.*

a.) Child

*% Child*

child(X, Y)*:-*

    parent(Y, X)*.*



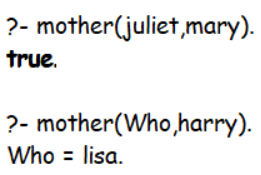
b.) Mother

*% Mother*

mother(X, Y)*:-*

    female(X),

    parent(X, Y)*.*



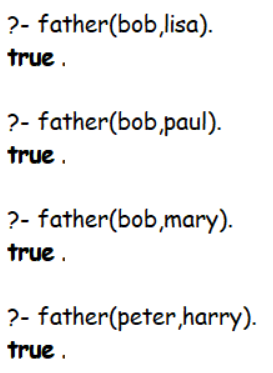
c.) Father

*% Father*

father(X, Y)*:-*

    male(X),

    parent(X, Y)*.*

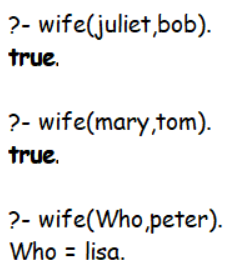


d.) Wife

*% Wife*

wife(X, Y)*:-*

    husband(Y, X)*.*



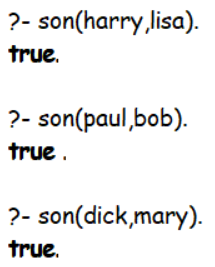
e.) Son

*% Son*

son(X, Y)*:-*

    male(X),

    parent(Y, X)*.*



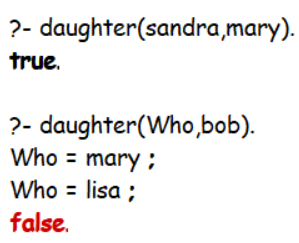
f.) Daughter

*% Daughter*

daughter(X, Y)*:-*

    female(X),

    parent(Y, X)*.*



g.) Brother

*% Brother*

brother(X,Y) *:-*

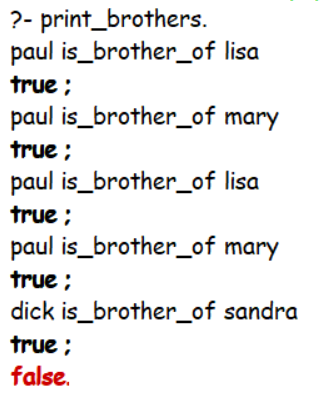
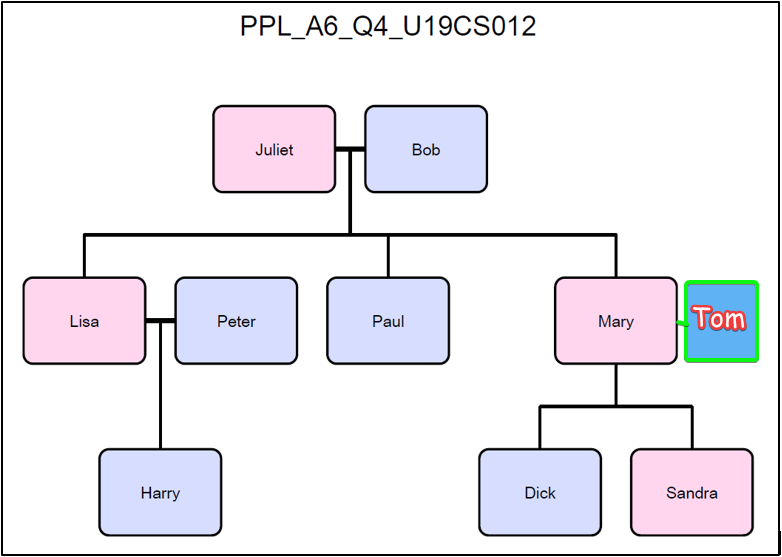
    male(X),

    parent(P,X),

    parent(P,Y),

    X \= Y*.*

print\_brothers *:-* brother(X, Y), format('~a is\_brother\_of ~a', [X, Y])*.*

h.) Sister

*% Sister*

sister(X,Y) *:-*

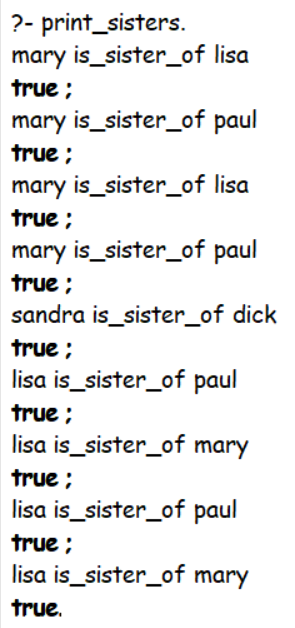
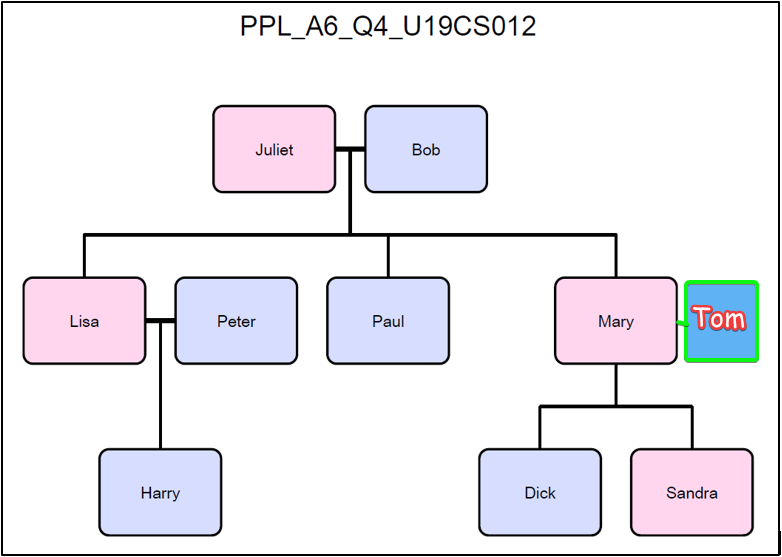
    female(X),

    parent(P,X),

    parent(P,Y),

    X \= Y*.*

print\_sisters *:-* sister(X, Y), format('~a is\_sister\_of ~a', [X, Y])*.*

i.) Uncle

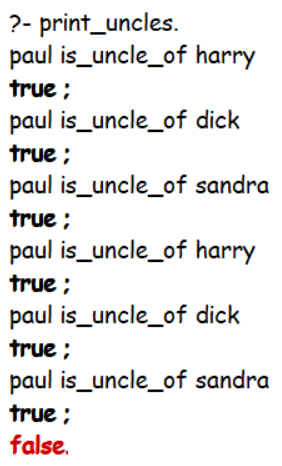
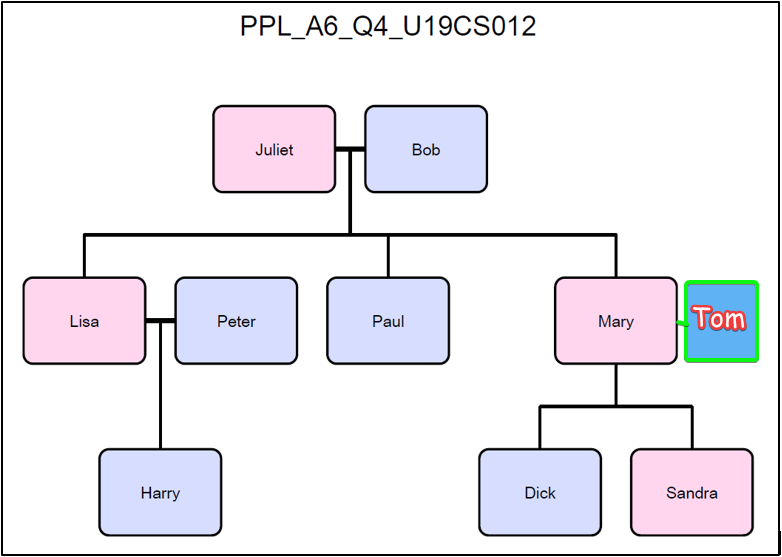
*% Uncle - X is an uncle of Y if X is the brother of some person, Z, and Z is a parent of Y*

uncle(X, Y) *:-*

    brother(X, Z),

    parent(Z, Y)*.*

print\_uncles *:-* uncle(X, Y), format('~a is\_uncle\_of ~a', [X, Y])*.*

j.) Aunt

*% Aunt - X is an aunt of Y if X is the sister of some person, P, and P is a parent of Y*

aunt(X, Y) *:-*

    sister(X, P),

    parent(P, Y)*.*

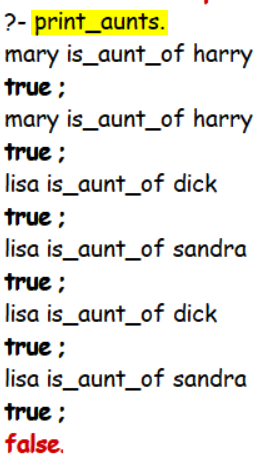
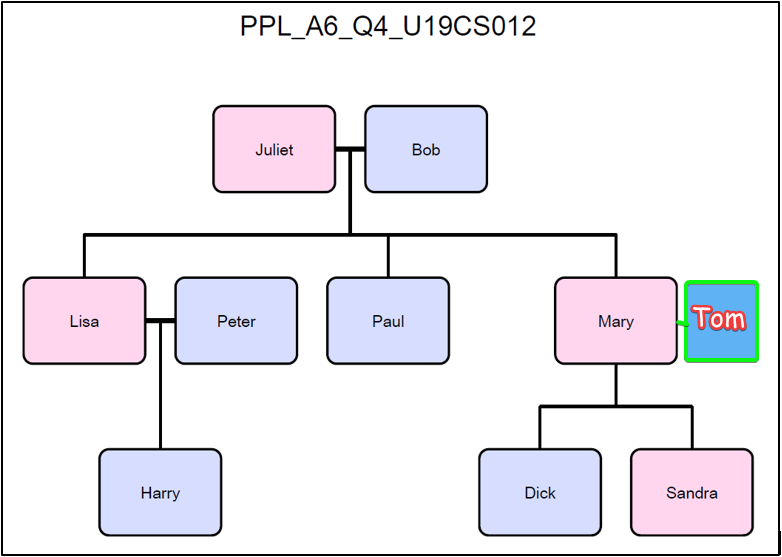
aunt(X, Y) *:-*

    parent(P, Y),

    brother(Uncle, P),

    wife(X, Uncle)*.*

print\_aunts *:-* aunt(X, Y), format('~a is\_aunt\_of ~a', [X, Y])*.*

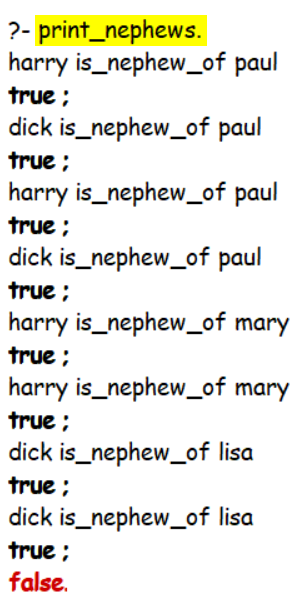
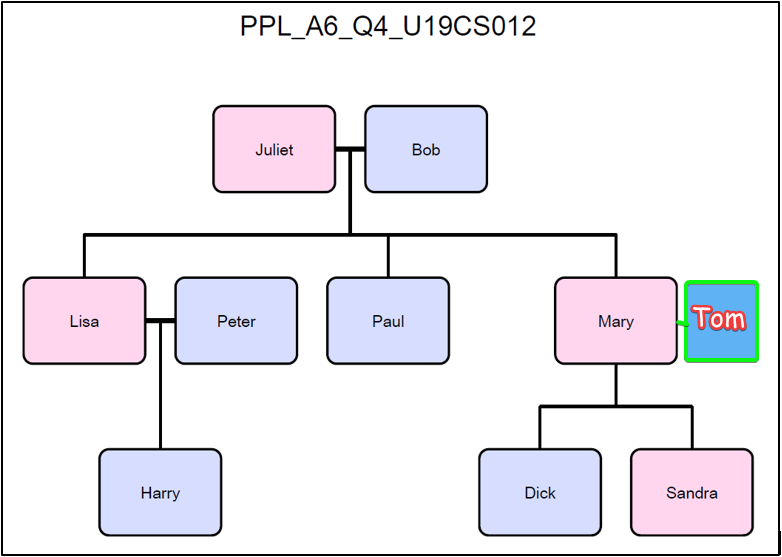
k.) Nephew

*% Nephew*

nephew(X, Y) *:-* uncle(Y, X), male(X)*.*

nephew(X, Y) *:-* aunt(Y, X), male(X)*.*

print\_nephews *:-* nephew(X, Y), format('~a is\_nephew\_of ~a', [X, Y])*.*

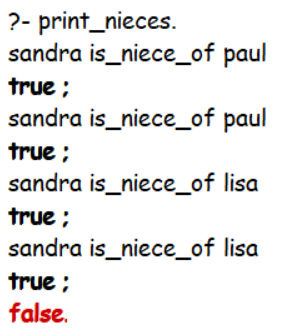
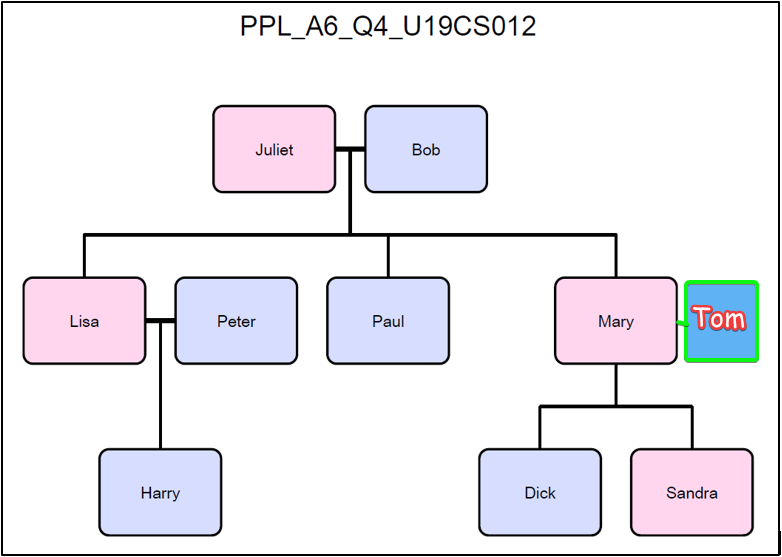
l.) Niece

*% Niece*

niece(X, Y) *:-* uncle(Y, X), female(X)*.*

niece(X, Y) *:-* aunt(Y, X), female(X)*.*

print\_nieces *:-* niece(X, Y), format('~a is\_niece\_of ~a', [X, Y])*.*

**SUBMITTED BY**: U19CS012

BHAGYA VINOD RANA