St. Francis Institute of Technology
Borivli (W), Mumbai 400103
Department of Information Technology

Experiment-6

1. Aim: a) To build knowledge base based on facts and rules
b) To implement logical expressions and codes using Prolog

2. Objectives: After performing the experiment, the students will be able to
- Download and install SWIProlog
- **Construct** knowledge base based on facts and rules
- Write logical expressions

3. Lab Objective Mapped: To understand, formulate and implement declarative programming paradigm through logic programming

4. Prerequisite: Nil

5. Requirements: The following are the requirements – • **Internet connection**
- **Laptop/desktop with Windows/Linux/MAC operating system**
- **SWI Prolog**

6. Pre-Experiment Theory:

Logic programming is a distinctive style of programming. It is based on collection of logical propositions and questions. There are only two components to any logic program: facts and rules.

- The program is a knowledge base of facts and a series of rules to be applied to knowledge base
- Programs are based on the techniques developed by logicians to form valid conclusions from available evidence (knowledge base)

Most widely used logic programming language is Prolog. The name stands for **Programming in Logic.**

FACTS

- Facts are statements about what is true about a problem, instead of instructions how to accomplish the solution.
- The Prolog system uses the facts to work out how to accomplish the solution by searching through the space of possible solutions.
  - It is defined by an identifier followed by an n-tuple of constants.
  - A relation identifier is referred to as a predicate
- When a tuple of values is in a relation we say the tuple satisfies the predicate.

Syntax of facts

- Names of relationship and objects must begin with a lower- case letter.

- Relationship is written *first* (typically the *predicate* of the sentence).
- *Objects* are written separated by commas and are enclosed by a pair of round brackets.
- The full stop character '.' must come at the end of a fact.

Example of Facts

```
valuable(gold).                    /*Gold is Valuable*/
owns(john, gold).                  /*John owns gold*/ father(john,
mary).          /*John is the father of Mary*/
gives (john, book,mary).           /*John gives the book to Mary*/
likes(john, Y), likes(Y, john).    /* John likes everybody and everybody likes John */
```

RULES

A **rule** is a predicate expression that uses logical implication (:-) to describe a relationship among facts. Thus a Prolog rule takes the form **left_hand_side :- right_hand_side .**

This sentence is interpreted as: *left_hand_side* ***if*** *right_hand_side*. The **left_hand_side** is restricted to a **single, positive, literal**, which means it must consist of a positive atomic expression. It cannot be negated and it cannot contain logical connectives.

- Rules specify under what conditions a tuple of values satisfies a predicate.
- The basic building block of a rule is called an *atom*
- Rules consists of Head of clause, body of clause and neck of clause

Examples of valid rules:

```
friends(X,Y) :- likes(X,Y),likes(Y,X).    /* X and Y are friends if they like each other */ hates(X,Y)
:- not(likes(X,Y)).          /* X hates Y if X does not like Y. */
```

Examples of invalid rules:

```
left_of(X,Y) :- right_of(Y,X)          /* Missing a period */ likes(X,Y),likes(Y,X)
:- friends(X,Y).     /* LHS is not a single literal */ not(likes(X,Y)) :- hates(X,Y).
/* LHS cannot be negated */
```

QUERIES

The Prolog interpreter responds to **queries** about the facts and rules represented in its database. The database is assumed to represent what is true about a particular problem domain. In making a query you are asking Prolog whether it can prove that your query is true. Whenever you run the Prolog interpreter, it will **prompt** you with **?-**.
Suppose your knowledge base consists of the following facts

```
food(burger). food(sandwich).
food(pizza).
lunch(sandwich). dinner(pizza).
```

Then if you put a query,

```
?- food(pizza).    /* Is pizza a food?
```

The answer to this query will be **True**

The standard logical operators in Prolog are as listed below in the Table 1.

Table 1: Prolog logical operators

Prolog Symbol	Read as	Logical Operations
,	AND	Conjunction
;	OR	Disjunction
not	NOT	Negation

Each of the logical operation is evaluated based on the truth table. The truth tables of the operations are given below-

NOT operation: If the input is true then the output will be false and vice versa

Table 2: Prolog NOT operation

Input	Output=not(input)
True	False
False	True

AND Operation: The output will be true if both the inputs are true

Table 3: Prolog AND operation

Input 1	Input 2	Output= Input 1, Input 2
False	False	False
False	True	False
True	False	False
True	True	True

OR Operation: The output will be true if any one of the inputs will be true

Table 4: Prolog OR operation

Input 1	Input 2	Output= Input 1; Input 2
False	False	False
False	True	True
True	False	True
True	True	True

NAND Operation- It's NOT of AND operation. The output will be false, if both the inputs will be true. The truth table is given below

Table 5: Prolog NAND operation

Input 1	Input 2	Output= not(Input 1, Input 2)
False	False	True
False	True	True
True	False	True
True	True	False

NOR Operation- It's NOT of OR operation. The output will be false, if any one of the inputs will be true. The truth table is given below

Table 6: Prolog NOR operation

Input 1	Input 2	Output= Input 1; Input 2
False	False	True
False	True	False
True	False	False

True	True	False

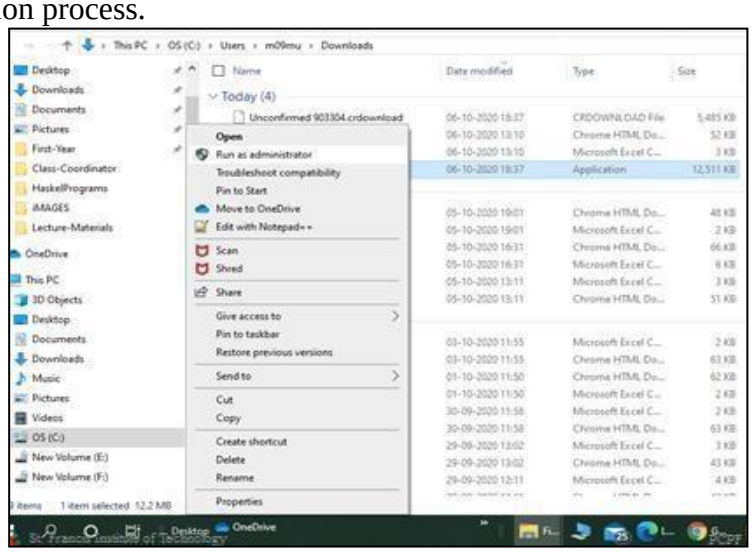
7. Lab Laboratory Exercise:

A. Procedure

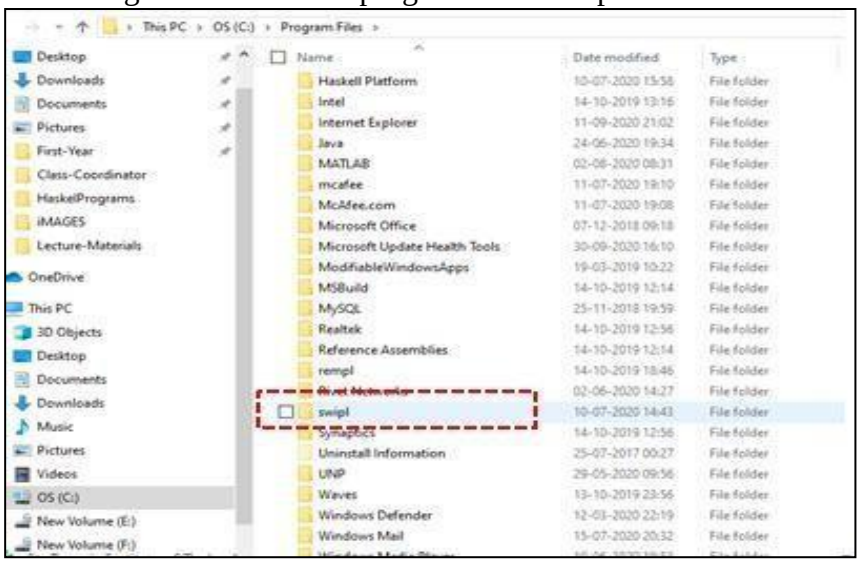
1. Visit https://www.swi-prolog.org/download/stable
2. Click on the highlighted box

3. Download the highlighted version

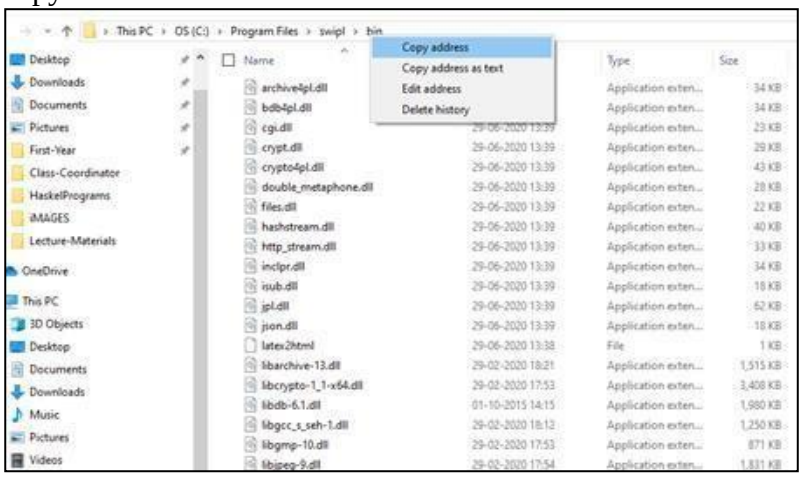
4. After successful download, install using 'Run as administrator'. This will start the installation process.



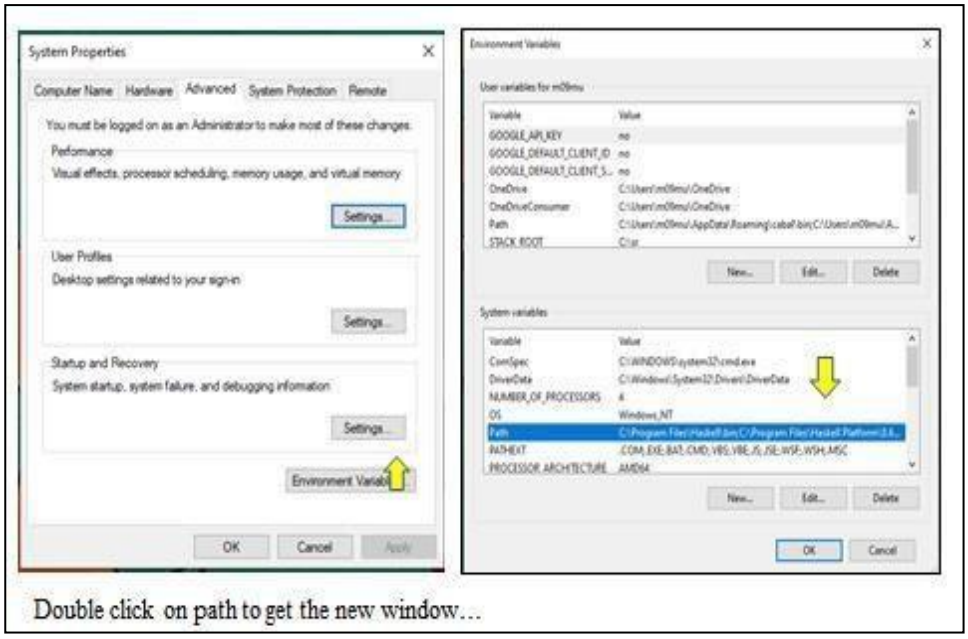
5. By default it will get installed in C:\program Files\Swipl folder



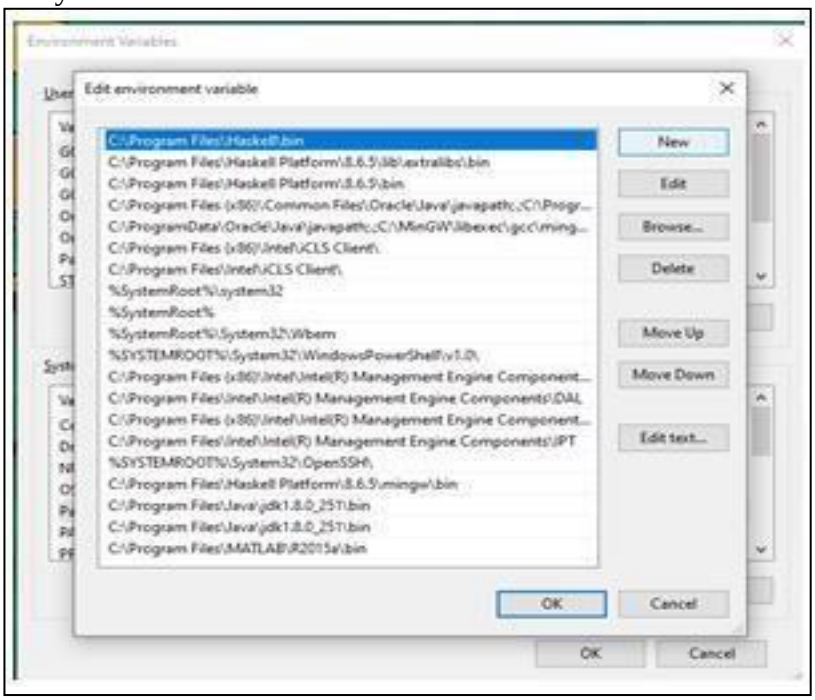
6. To set the path copy the entire address till bin



7. To set the path , go to system variables (under advanced system settings) and select the ‘path’ variable



8. Click on NEW and paste the entire copied path till the bin to set the path. Your path is now set successfully



9. On the desktop, double click on the SWI-Prolog Icon to open the prompt

B. Program Code

1. WAP in prolog to create knowledge base using the following facts

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

For the above knowledge base, fire the following queries and list the outputs ?-

woman(mia).

?- woman(jody).

?- woman(yolanda).

?- playsAirGuitar(jody).

?- playsAirGuitar(mia).

2. WAP in prolog to create knowledge base using the following facts and rules
happy(yolanda). listens2Music(mia).
listens2Music(yolanda):- happy(yolanda).
playsAirGuitar(mia):- listens2Music(mia).
playsAirGuitar(yolanda):-listens2Music(yolanda)
For the above knowledge base, fire the following queries- ?-happy(yolanda).

?-happy(mia).

?-listens2Music(mia).

?-listens2Music(yolanda).

?-happy(X).

?-playsAirGuitar(Y)

3. Write a prolog to create knowledge base using the following facts and rules

studies(charlie, csc135). studies(olivia, csc135).

studies(jack, csc131). studies(arthur, csc134).

teaches(kirke, csc135). teaches(collins,

csc131). teaches(collins, csc171).

teaches(juniper, csc134). professor(X, Y) :-

teaches(X, C), studies(Y, C).

For the above knowledge base, answer the following queries- ?-

studies(charlie, What).

?- professor(kirke, Students).

Analyse and explain the output

4 WAP in Prolog to implement the truth tables of the logical operations-NOT, AND, OR, NAND and NOR operations.

5 Implement the following logical operations • 6*6=:=36;10=8+3.

a sqrt(36)+4=:=5*11-45.
b 10=\=8+3.
c 111=\=8+3.
d 6+4==3+7.
e 6<3;7 is 5+2. □ not(111=\=8+3).

- 111=\=8+3,11=\=3. 111=\=8+3;11=\=3.
- sqrt(36)+4=:=5*11-45;false. sqrt(36)+4=:=5*11-45,false.
- not(6<3;7 is 5+2).
- not(not(6<3;7 is 5+2)).
- not(not(6<3;7 is 5+2)),true.
- not(not(6<3;7 is 5+2)),false.
- not(not(6<3;7 is 5+2));false. not(not(6<3;7 is 5+2));true.

8. Post Experimental Exercise-

A. Questions:

1. Define the terms- (i) atom (ii) variable. Give examples
2. For the above two knowledge, make a table to enlist the English meaning of the facts and rules
3. Explain the significance of the operators **‘,’** and **‘:-‘** operators w.r.t. to Prolog

4 WAP to create the following knowledge base loves(vincent,mia). loves(marsellus,mia). loves(pumpkin,honey_bunny). loves(honey_bunny,pumpkin). jealous(X,Y):- loves(X,Z), loves(Y,Z). Also generate five different queries to get results

5. Write a program in Prolog to include the following facts and rules in the knowledge base dog(fido). dog(kitty). cat(sweety). cat(micky). animal(X):-dog(X). noanimal(Y):cat(Y).

   On this knowledge base, implement the following queries and record your answers not(dog(fido)). not(dog(kitty)). not(cat(sweety)). not(cat(micky)). not(dog(fiddo)). not(cat(sweetty)). not(cat(sweety)),not(cat(sweetty)). not(cat(sweety));not(cat(miccy)). not(cat(sweety));not(cat(micky)).

6 Describe all the logical operators in prolog.

7 On the knowledge base of program 1, implement the NAND and NOR operation

B. Results/Observations/Program output:

Present the program input/output results if any and comment on the same.

C. Conclusion:

1. Write what was performed in the experiment

2. Write which tools you used to perform the experiment

3. Write what you inferred from the output obtained

D. References:

[1] Michael L Scott, “Programming Language Pragmatics”, Third edition, Elsevier publication
[2] Max Bramer, “ Logic Programming with Prolog”, Springer, 2005