SHIVA KANT MISHRA   1900321290057

Year/Semester: 4th /VII   Section: CEIT

Experiment-1 (A): Study of Prolog.

OBJECTIVE: Prolog programming basics

What is Prolog?
- Prolog stands for *programming in logic*.
- Prolog is a declarative language, which means that a program consists of data based on the facts and rules (Logical relationship) rather than computing how to find a solution.
- A logical relationship describes the relationships which hold for the given application.
- To obtain the solution, the user asks a question rather than running a program. When a user asks a question, then to determine the answer, the run time system searches through the database of facts and rules.
- Prolog is a declarative language that means we can specify what problem we want to solve rather than how to solve it.
- Prolog is used in some areas like database, natural language processing, artificial intelligence, but it is pretty useless in some areas like a numerical algorithm or instance graphics.
- In artificial intelligence applications, prolog is used. The artificial intelligence applications can be automated reasoning systems, natural language interfaces, and expert systems. The expert system consists of an interface engine and a database of facts. The prolog's run time system provides the service of an interface engine.
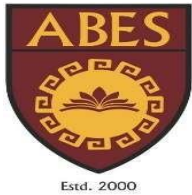
Applications of Prolog
- Specification Language
- Robot Planning
- Natural language understanding
- Machine Learning
- Problem Solving
- Intelligent Database retrieval
- Expert System
- Automated Reasoning

Starting Prolog

SHIVA KANT MISHRA   1900321290057
Year/Semester: 4th /VII   Section: CEIT
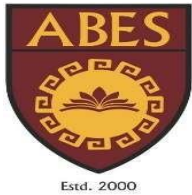
- Prolog system is straightforward.
- Prolog will produce a number of lines of headings in the starting, which is followed by a line. It contains just
- ?-
- The above symbol shows the system prompt. The prompt is used to show that the Prolog system is ready to specify one or more goals of sequence to the user.
- Using a full stop, we can terminate the sequence of goals.
- **For example:**
- **?- write('Welcome to AI lab Class'),nl,write('Example of Prolog'),nl.**
- **nl** indicates 'start a new line'. When we press 'return' key, the above line will show the effect like this:
  Welcome to AI Lab
  Class Example of
  Prolog yes
- **?- prompt** shows the sequence of goal which is entered by the user. The user will not type the prompt.
- Prolog system will automatically generate this prompt. It means that it is ready to receive a sequence of goals.
- To show that the goals have succeeded, we will output yes.
- **'Query'** is a sequence of one or more goals.

Prolog Programs

- To write a Prolog program, firstly, the user has to write a program which is written in the Prolog language, load that program, and then specify a sequence of one or more goals at the prompt.
- To create a program in Prolog, the simple way is to type it into the text editor and then save it as a text file like prolog1.pl.
- The following example shows a simple program of Prolog. The program contains *three components, which are known as clauses. Each clause is terminated using a full stop.* dog(rottweiler).   cat(munchkin).
  animal(A) :- cat(A).
- Using the built-in predicate '**consult'**, the above program can be loaded in the Prolog system.

- **?-consult('prolog1.pl').**

Subject Code: KCS-751A   Subject Name: Artificial Intelligence Lab

SHIVA KANT MISHRA   1900321290057
Year/Semester: 4th /VII   Section: CEIT

- This shows that prolog1.pl file exists, and the prolog program is systemically correct, which means it has valid clauses, the goal will succeed, and to confirm that the program has been correctly read, it produces one or more lines of output. e.g., **?-**
 # 0.00 seconds to consult prolog1.pl
?-
- The alternative of 'consult' is 'Load', which will exist on the menu option if the Prolog system has a graphical user interface.
- When the program is loaded, the clause will be placed in a storage area, and that storage area is known as the Prolog database. In response to the system prompt, specify a sequence of goals, and it will cause Prolog to search for and use the clauses necessary to evaluate the goals.

Terminology
In the following program, three lines show the
clauses. dog(rottweiler). cat(munchkin).
animal(A) :- cat(A).
Using the full stop, each clause will be terminated. Prolog programs have a sequence of clauses. Facts or rules are described by these clauses.
Example of **facts** is **dog(rottweiler)** and **cat(munchkin)**. They mean that '**rottweiler** is a dog' and '**munchkin** is a cat'.

Dog is called a predicate. Dog contains one argument. Word '**rottweiler'** enclosed in bracket( ). Rottweiler is called an atom.
The example of rule is the final line of the program.
animal(A) :- dog(A).
The colon(:-) character will be read as 'if'. Here A is a variable, and it represents any value. In a natural way, the rule can be read as "If A is an animal, then A is a dog".

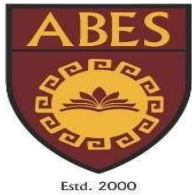The above clause shows that the **rottweiler** is an animal.
Such deduction can also make by Prolog:
?- animal(rottweiler). yes
To imply that munchkin is an animal, there is no evidence of this.
?- animal(munchkin).
no

Subject Code: KCS-751A   Subject Name: Artificial Intelligence Lab

SHIVA KANT MISHRA   1900321290057
Year/Semester: 4th /VII   Section: CEIT

Experiment-1 (B)

OBJECTIVE: Write simple fact for the statements using PROLOG.

Facts A fact is like a predicate expression. It is used to provide a declarative statement about the problem. In a Prolog expression, when a variable occurs, it is assumed to be universally quantified. Facts are specified in the form of the **head**. Head is known as the clause head. It will take in the same way as the goal entered at the prompt by the user.   cat(bengal).   /* bengal is a cat */   dog(rottweiler). /* rottweiler is a dog */   likes(Jolie, Kevin).   /* Jolie likes Kevin */   likes(A, Kevin).   /* Everyone likes Kevin */   likes(Jolie, B).   /* Jolie likes everybody */   likes(B, Jolie), likes(Jolie, B).   /* Everybody likes Jolie and Jolie likes everybody * /

likes(Jolie, Kevin); likes(Jolie, Ray). /* Jolie likes Kevin or Jolie likes Ray */

not(likes(Jolie, pasta)).   /* Jolie does not like pasta */

Queries

In Prolog, the query is the action of asking the program about the information which is available within its database. When a Prolog program is loaded, we will get the query prompt,

?-

After this, we can ask about the information to the run time system. Using the above simple database, we can ask a question to the program like ?- 'It is sunny'.

and it will give the answer

yes   ?-

The system responds to the query with yes if the database information is consistent to answer the query. Using the available database information, we can also check that the program is capable of proving the query true. No indicates that the fact is not deducible based on the available information.
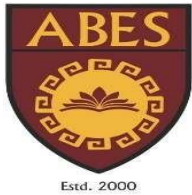
The system answers no to the query if the database does not have sufficient information. ?- 'It is cold'.

no

  ?-

Write simple fact for following: a.

Ram likes mango.

b. Seema is a girl.

SHIVA KANT MISHRA   1900321290057

Year/Semester: 4th /VII   Section: CEIT

c. Bill likes Cindy.
d. Rose is red.
e. John owns gold.

Program:

```
File Edit Format View Help
likes(ram ,mango).
girl(seema).
red(rose).
likes(bill ,cindy).
owns(john ,gold).
```

Output:

```
?- likes(ram,What).
What = mango.

?- red(What).
What = rose.

?- owns(Who,What).
Who = john,
What = gold.

?- girl(Who).
Who = seema.
```

Example 2.
Statements:
1. The Cakes are delicious.
2. The Pickles are delicious.
3. The Pickles are spicy.
4. Priya relishes coffee.
5. Priya likes food if they are delicious.
6. Prakash likes food if they are spicy and delicious.

Statements in Prolog:
1. delicious(cakes).
2. delicious(pickles).
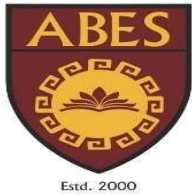
SHIVA KANT MISHRA   1900321290057

Year/Semester: 4th /VII   Section: CEIT

3. spicy(pickles).
4. relishes(priya, coffee).
5. likes(priya, Food) if delicious(Food). %Here Food is a variable
   6.   likes(prakash,Food) if spicy(Food) and delicious(Food).

Program Clauses:

```
File  Edit  Format  View  Help
delicious(cakes).
delicious(pickles).
spicy(pickles).
relishes(priya, coffee).
% here Food is a variable
likes(priya, Food):-delicious(Food).
likes(prakash, Food):- spicy(Food), delicious(Food).
```

Goal 1- Which food items are delicious.

Goal 2- What food does priya like.

Goal 3 what food does Prakash like.

Screenshot Output-

```
?- delicious(Food).
Food = cakes .

?- likes(priya,Food).
Food = cakes .

?- likes(prakash,Food).
Food = pickles.
```
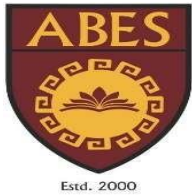
SHIVA KANT MISHRA   1900321290057
Year/Semester: 4th /VII   Section: CEIT

Experiment-2 (A)

OBJECTIVE: Write predicates One converts centigrade temperatures to Fahrenheit, the other checks if a temperature is below freezing.
Formula for Centigrade (C) temperatures to Fahrenheit (F) - F
= C * 9 / 5 + 32

Rule
Centigrade to Fahrenheit (c_to_f)F is C * 9 / 5 + 32
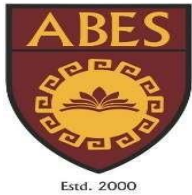
Program-

```
File  Edit  Format  View  Help
celsius_to_farenhiet(C,F):- F is C*9/5+32.
freezing(F):- F<32.
|
```

Goal to find Fahrenheit temperature and freezing point

Output:

```
?- celsius_to_farenhiet(32,F).
F = 89.6.


?- freezing(31).
true.
```

Subject Code: KCS-751A   Subject Name: Artificial Intelligence Lab

SHIVA KANT MISHRA   1900321290057
Year/Semester: 4th /VII   Section: CEIT

Experiment-2 (B)

**OBJECTIVE: Write a program to solve the Monkey Banana problem.**

Monkey wants the bananas but he can't reach them. What
shall he do? The monkey is in the room.
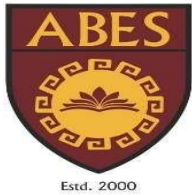Suspended from the roof, just out of his reach, is a bunch of bananas.
In the corner of the room is a box. The monkey desperately wants to grasp bananas.
After several unsuccessful attempts to reach the bananas:

1. The monkey walks to the box.
2. Pushes it under the bananas.
3. Climb on the box.
4. Picks the banana & eats them.
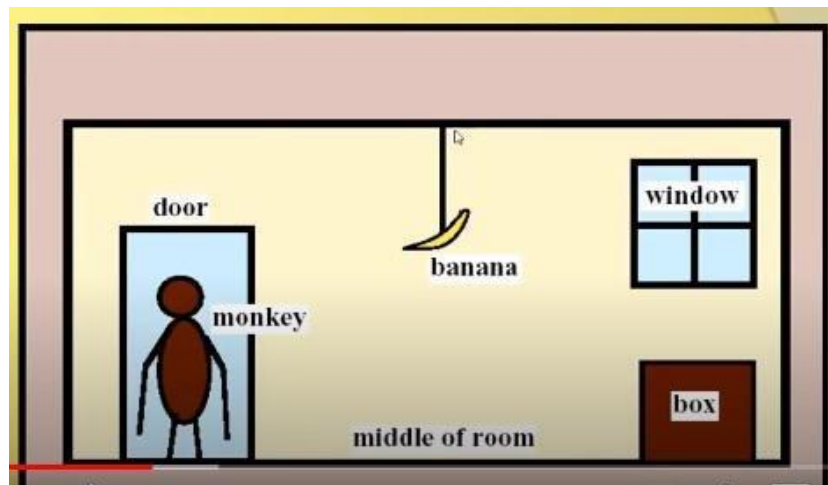
Program:

SHIVA KANT MISHRA   1900321290057

Year/Semester: 4th /VII   Section: CEIT

```
File  Edit  Format  View  Help
on(floor,monkey).
on(floor,box).
in(room,monkey).
in(room,box).
at(ceiling,bnana).

strong(monkey).
grasp(monkey).
climb(monkey,box).
push(monkey,box):-
    strong(monkey).

under(banana,box):-
    push(monkey,box).
canreach(banana,monkey):-
    at(floor,banana);
    at(ceiling,banana);
    under(banana,box).
canget(banana,monkey):-
    canreach(banana,monkey),grasp(monkey).
```
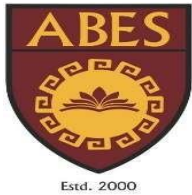
OUTPUT:

```
?- canreach(banana,monkey).
true.

?- trace.
true.
```

SHIVA KANT MISHRA   1900321290057
Year/Semester: 4th /VII   Section: CEIT

```
[trace]  ?- canget(banana,monkey).
  Call: (10) canget(banana, monkey) ? creep
  Call: (11) canreach(banana, monkey) ? creep
  Call: (12) at(floor, banana) ? creep
  Fail: (12) at(floor, banana) ? creep
  Redo: (11) canreach(banana, monkey) ? creep
  Call: (12) at(ceiling, banana) ? creep
  Fail: (12) at(ceiling, banana) ? creep
  Redo: (11) canreach(banana, monkey) ? creep
  Call: (12) under(banana, box) ? creep
  Call: (13) push(monkey, box) ? creep
  Call: (14) strong(monkey) ? creep
  Exit: (14) strong(monkey) ? creep
  Exit: (13) push(monkey, box) ? creep
  Exit: (12) under(banana, box) ? creep
  Exit: (11) canreach(banana, monkey) ? creep
  Call: (11) grasp(monkey) ? creep
  Exit: (11) grasp(monkey) ? creep
  Exit: (10) canget(banana, monkey) ? creep
true.
```
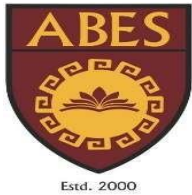
SHIVA KANT MISHRA   1900321290057
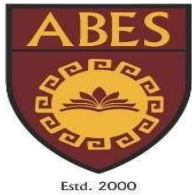Year/Semester: 4th /VII   Section: CEIT

Experiment-3

OBJECTIVE: Write a program to design medical diagnosis expert system in SWI
Prolog. The system must work for diagnosis of following diseases: -
a)   measles
b)   germanmeasles
c)   Flu
d)   commoncold
e)   mumps
f).   chickenpox

SHIVA KANT MISHRA   1900321290057
Year/Semester: 4th /VII   Section: CEIT

Program Code:

```
File Edit Format View Help
symptom(Ayush, fever)
symptom (Ayush, headache).
symptom(Ayush, runnynose).
symptom(Ayush, rash).

symptom(amit, headache).
symptom(amit, runnynose).
symptom (amit, snuzing).
symptom(amit, chills).
symptom (amit, sorethrought).
symptom(aman, runnynose).
symptom(aman, snuzing).
symptom(aman, cough).

symptom(rajesh, fever).
symptom(rajesh, rash).
symptom(rajesh, bodyache).
symptom(rajesh, swallenglands).

symptom(guarav, fever).
symptom(guarav, rash).
symptom(guarav, headache).
symptom(guarav, bodyache).
symptom(guarav, chills).
symptom(guarav, sorethrought).
symptom(guarav, cough).
symptom(guarav, conjunctive).
symptom(guarav, runnynose).
```

```
hypothesis(Patient,measles) :-
        symptom(Patient,fever),symptom(Patient,cough), symptom(Patient,conjunctive), symptom(Patient,runnynose), symptom(Patient,rash).
hypothesis(Patient,germanmeasles):-
        symptom(Patient,fever),symptom(Patient,headache),symptom(Patient,runnynose),symptom(Patient,rash).
hypothesis(Patient,flu):-
        symptom(Patient,fever),symptom(Patient,headache),symptom(Patient,bodyache),symptom(Patient,chills),
        symptom(Patient,sorethrought),symptom(Patient,cough), symptom(Patient,conjunctive),symptom(Patient,conjunctive),symptom(Patient,runnynose).
hypothesis(Patient,commoncold):-
        symptom(Patient,headache),symptom(Patient,runnynose),symptom(Patient,snuzing),symptom(Patient,chills),symptom(Patient,sorethrought).
hypothesis(Patient,mumps):-
        symptom(Patient,fever),symptom(Patient,swallenglands).
hypothesis(Patient,chikenpox):-
        symptom(Patient,fever),symptom(Patient,rash),symptom(Patient,bodyache),symptom(Patient,swallenglands).
```
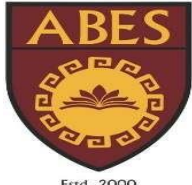
OUTPUT:

```
?- hypothesis(Patient,measles).
Patient = guarav.

?- hypothesis(Who,chikenpox).
Who = rajesh .

?- hypothesis(Who,commoncold).
Who = amit .
```

SHIVA KANT MISHRA   1900321290057
Year/Semester: 4th /VII   Section: CEIT