**Starting Turbo prolog::**

Z:\> mount c c:\tprolog

Z:\> c:

c:\> prolog

Then the user-friendly editor appears as follows:-

|  |  |  |
| --- | --- | --- |
| **Files Edit Run Compile Options Setup** | | |
| **Editor**  **Line1 Col1 Indent Insert Work.pro** | | **----Dialog-----** |
| **--------Message---------** | **--------Trace----------** | |
|  |  |  |

**Introduction to Prolog ::**

There are only three basic constructs in Prolog: facts, rules, and queries. A collection of facts and rules is called a knowledge base (or a database) and Prolog programming is all about writing knowledge bases.

**Facts :**:

Facts are those statements that state the objects or describe the relationship between objects. For an instance when we say *john likes piano*, we are showing the 'like' relationship between two objects 'john and piano' and in prolog this fact can be written as *likes(john,piano)*.

**Example 1:**:   
music\_student(john).       // Read as : john is a music student                       
 likes(John,books)            // Read as : john likes books  
 gives(john, chocolate).    // Read as : john gives chocolate

SAMPLE TEST PROGRAM

|  |  |  |
| --- | --- | --- |
| **Run Compile Edit Options Files Setup Quit** | | |
| **Editor**  **Line1 Col1 Indent Insert Work.pro**  **/\* SAMPLE TEST PROGRAM \*/**  **predicates**  **likes(symbol , symbol)**  **clauses**  **likes(frank,muna).**  **likes(ahmed,book).** | | **----Dialog-----** |
| **--------Message---------** | **--------Trace----------** | |
|  |  |  |

**F1:Help F3:Search F4:Subst F5:Copy F6:Move F7:del F8:ExtEdit F9:ExtCopy F10:End**

**How to query ::**

Once you have entered the facts in a program you can ask prolog about it. An example program can be :

eats(fred,oranges). /\* 'Fred eats oranges' \*/

eats(tony,apple). /\* 'Tony eats apple' \*/

eats(john,apple). /\* 'John eats apple' \*/

If we now ask some queries we would get the followings things :

?- eats(fred,oranges).

/\* does this match anything in the database? \*/

yes

/\* yes, that matches the first clause \*/

?- eats(john,apple).

yes

?- eats(mike,apple).

no

/\* there is no relation between mike and apple \*/

**Examples 2:**:

lectures(ahmed, ai).

lectures(john, databases).

female(alison).

age(alison, 29).

office(alison, s134).

animal(lion).

?- lectures(alison, ai).

``yes''.

?- lectures(alison, databases).

no

## Variables ::

If we have the following program : eats(fred,oranges).

How can we ask something like ''what does Fred eat ?'' How do we ask what fred eats ? We could write something like this :

?- eats(fred,what)

But Prolog will say no. The reason is that Prolog can't find the relation eats(fred,what) in the database. In this case we have to use a variable

X /\* a single capital letter \*/

Variable /\* a word beginning with an upper case letter \*/

Now that we know how to use a variables, we can ask the same question as before using the variable

?- eats(fred,What)

What=oranges

yes