Lab#2

***Object:*** *Understanding variables and rules in PROLOG.*

**2.1 Rules**

Till now, we added only facts to our program but the real power of Prolog is in rules. While facts state the relation explicitly, rules define the relation in a more general way. Each rule has its head - name of the defined relation, and its body - a real definition of the relation.

A rule can be viewed as an extension of a fact with added conditions that also have to be satisfied for it to be true. It consists of two parts. The first part is similar to a fact (a predicate with arguments). The second part consists of other clauses (facts or rules which are separated by commas) which must all be true for the rule itself to be true. These two parts are separated by “:-” You may interpret this operator as "if" in English.

A rule is an expression that indicates that the truth of a particular fact depends upon one or more other facts. The syntax can be:

**predicate1(arg1, arg2, ... argN):- predicate2(arg1, arg2, ... argN), predicate3(arg1, arg2, ... argN),……… ,predicateN(arg1, arg2, ... argN).**

This can be read as predicate1 is true only if predicate2 to predicateN are true.OrIt can be: **predicate1(arg1, arg2, ... argN):- predicate2(arg1, arg2, ... argN); predicate3(arg1, arg2, ... argN);……… ;predicateN(arg1, arg2, ... argN).**

This can be read as predicate1 is true if any one of predicate2 to predicate N is true (as ; represents OR in Prolog).The left hand side before :- of a rule 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.

***Here are some examples of rules:***

***Example 1:***Person1 is mother of Person2 if Person1 is female and Person1 is parent of person2.

**mother(Person1, Person2) :- female(Person1) , parent(Person1,Person2).**

***Example 2:***Person1 is friend of Person2 if Person1 knows Person2 and Person1 likes Person2.

**friend(Person1, Person2) :- knows(Person1, Person2) , likes(Person1, Person2).**

***Example 3:***A mobile is off if it is out of battery or it is shutdown.

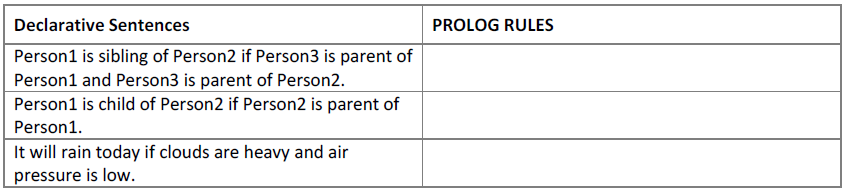
**off(mobile) :- outofbattery(mobile) ;shutdown(mobile).**

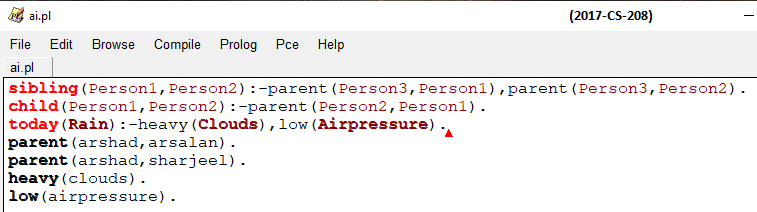
**2.2 Symbols used in rules**

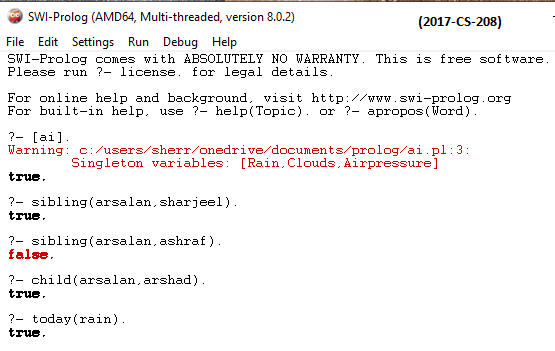
|  |  |
| --- | --- |
| **Symbol** | **Read as** |
| **:-** | **If or implies that** |
| **,** | **And** |
| **;** | **Or** |
| **not** | **Not** |

***Tasks to do:***

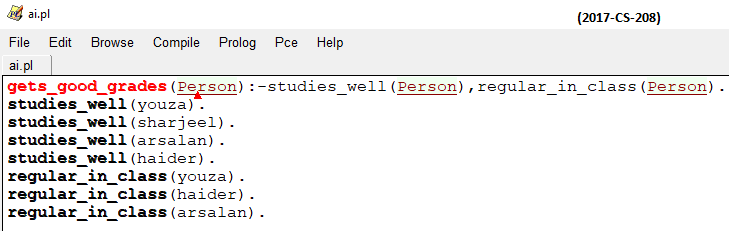
**1. Write down the PROLOG Rules for the following sentences:**

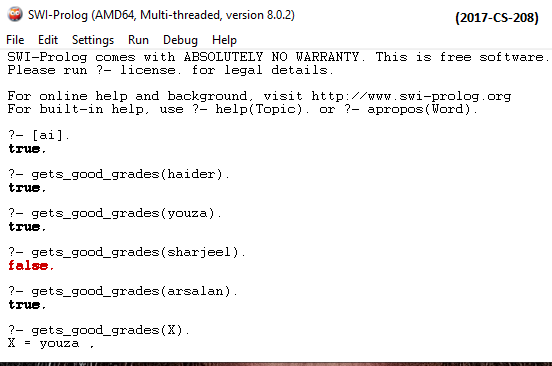


******

******

**2. A person gets good grades if he studies well and is regular in class. Write program in prolog to check whether a given person gets good grade or not?**

****

****

**3. Here are some declarative sentences convert them into PROLOG facts and answer following questions.**

Kashif likes music.

Kashif likes cricket.

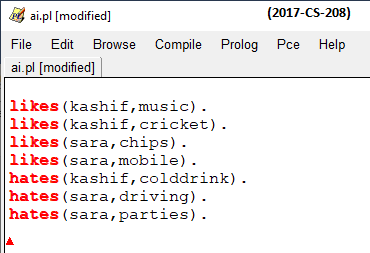
Sara likes chips.

Sara likes mobile.

Kashif hates cold drinks.

Sara hates driving.

Sara hates parties.



***Questions (GOALS):***

Does Sara like chips?

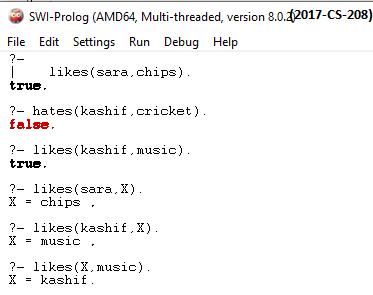
Does Kashif hate cricket?

Does Kashif like music?

What does kashif like?

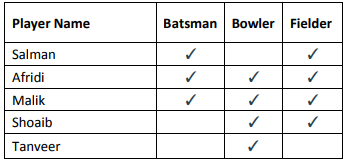
What does Sara like?

Who likes music?

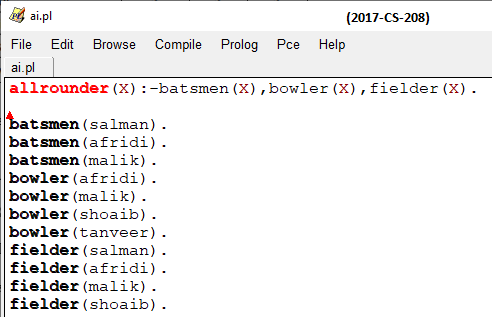
****

**4. Cricket case study**

**There is a cricket team of 5 players; capabilities (properties) of each of the player are given in following table*.***

****

**Write the facts from the above table, create the rule for all Rounder and run following goals in PROLOG.**

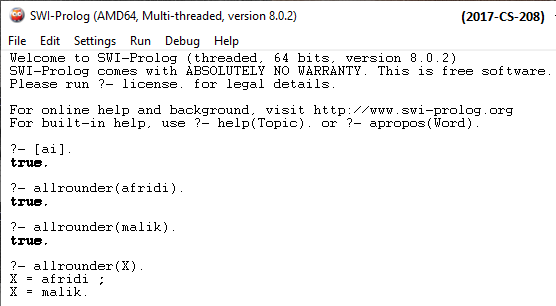
****

**QUESTIONS:**

**1. Is Afridi all rounder?**

**2. Is Shoaib all rounder?**

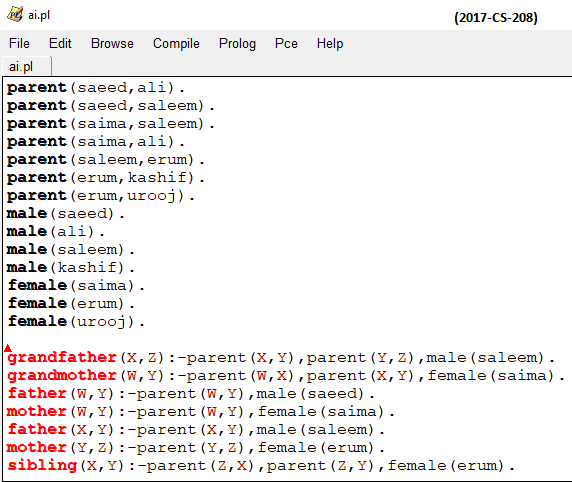
**3. Who are all rounders in the team?**

****

**5. Case Study: Family Tree**

Consider the following family tree shown in figure below



****

**Questions:**

**Is saleem father of ali?**

**Is kashif brother of urooj?**

**Who is father of kashif?**

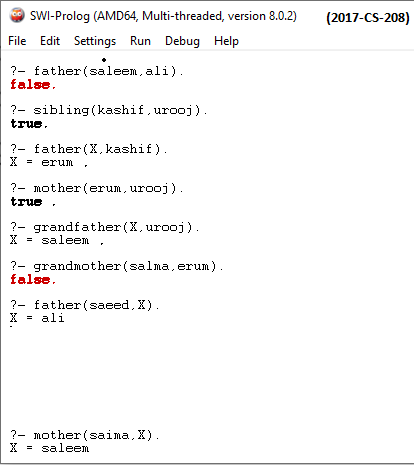
**Is erum mother of urooj?**

**Who is grandfather of urooj?**

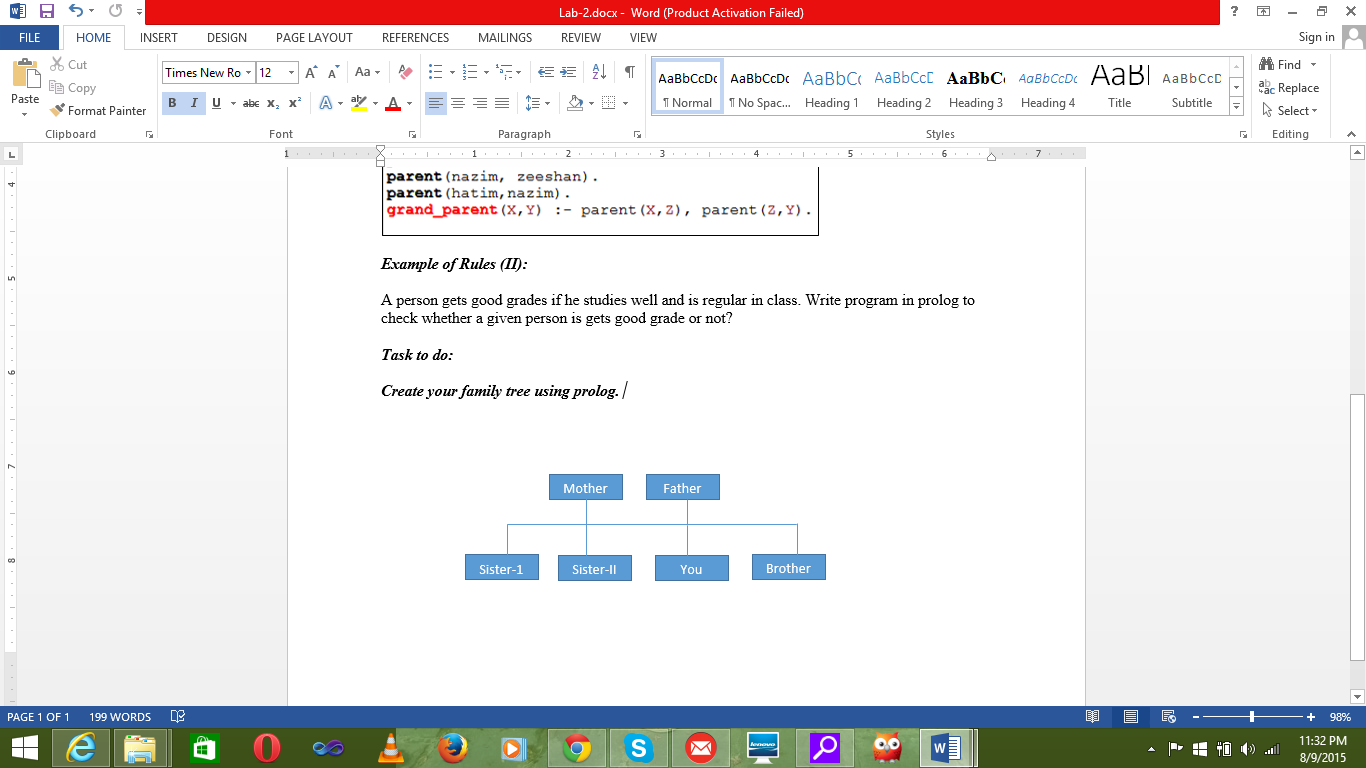
**Is saima grandmother of erum?**

**Saeed is father of whom?**

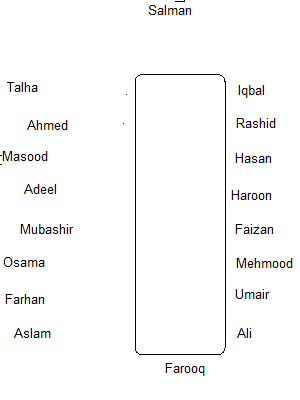
**Salma is mother of whom?**

****

**6. Create your family tree using prolog.**



**7. The following picture shows who is sitting at the lunch table.**



Define the predicate sits\_right\_of to represent who is sitting right of whom.sits\_right\_of(X,Y) should be true if X is to the right of Y.

Based on this knowledge base, formulate the rules defining the following predicates:

**(i) left\_of**:**sits\_left\_of(X,Y)**should be true if**X**is to the left of**Y**.

**(ii) are\_neighbors\_of**:**are\_neighbors\_of(X,Y,Z)**should be true if**X**is to the left of**Y or  Z** is to the right of**Y**.

**(iii) Y is in centre of X and Y(X,Y,Z)**should be true if**X**is to the left of **Y** and **Z** is to the right of Y.

Test your implementation by asking queries. For example:

1. Is Osama to the right of Farhan?
2. Is Osama to the right of Mubashir?
3. Who is to the right of Umair?
4. Who is sitting at the table?
5. Who is sitting between Mubashir and Masood?

**8. Selection criteria using rules.**

* starter(soup).
* starter(salad).
* main\_course(rice).
* main\_course(pizza).
* main\_course(burger).
* desert(custard).
* desert(ice\_cream).

***Task To do:***

* Create rules for hungry, very\_hungry and on\_diet.
* For hungry only starter and main course, for very\_hungry starter, main course and desert, for on\_diet it displays only starters.