Assignment 1 Documentation

Source code with comments/documentation:-

(The comments are written in bold italics and start with % sign)

% course fact contains --> Name of course, Course department, Area of interest of course, Absolute/Relative grading, pre_reqs.

% Here we define which facts are dynamic

```
:- dynamic department/1, interest/1, grading/1, pre_reqs/1.
```

% Machine Learning Courses(CSE courses)

```
course('Machine Learning','CSE','Machine Learning','relative',['MTH100','CSE101']).
course('Natural Language Processing','CSE','Machine
Learning','absolute',['MTH201','CSE101']).
course('Artificial Intelligence','CSE','Machine Learning','relative',['CSE102','CSE121']).
```

% Networking Courses(CSE courses)

```
course('Computer Networks','CSE','Networking','relative',['CSE101','CSE222','CSE232']).
course('Network Security','CSE','Networking','absolute',['CSE232','CSE231']).
course('Computer Security','CSE','Networking','relative',['CSE232']).
```

% Databases Courses(CSE courses)

```
course('Information
Retrieval','CSE','Databases','relative',['CSE102','CSE201','CSE202']).
course('DBMS','CSE','Databases','relative',['CSE102']).
course('Information Integration and
Applications','CSE','Databases','absolute',['CSE202']).
```

% Algorithms Courses(CSE courses)

```
course('Algorithm Design and Analysis','CSE','Algorithms','relative',['CSE102']).
course('Introduction to Programming','CSE','Algorithms','relative',[]).
course('Modern Algorithm Design','CSE','Algorithms','absolute',['CSE222']).
course('Approximation Algorithms','CSE','Algorithms','relative',['CSE222']).
```

% Biology Courses(Non-CSE courses)

```
course('Computational Gastronomy','Non-CSE','Biology','relative',[]).
course('Computing for Medicine','Non-CSE','Biology','relative',[]).
course('Machine Learning for Biomedical Applications','Non-CSE','Biology','relative',[]).
```

% Design Courses(Non-CSE courses)

```
course('Introduction to Animation and Graphics','Non-CSE','Design','absolute',[]).
course('3D Animation Film Making','Non-CSE','Design','relative',[]).
course('Affective
Computing','Non-CSE','Design','relative',['CSE101','CSE102','CSE201']).

% Social Science Courses(Non-CSE courses)
course('Neuroscience of Decision Making','Non-CSE','Social Science','absolute',[]).
course('Urban Space and Political Power','Non-CSE','Social Science','relative',[]).
course('Environmental Sciences','Non-CSE','Social Sciences','absolute',[]).

course('Digital Signal Processing','Non-CSE','Electronics','absolute',['ECE250']).
course('Reinforcement Learning','Non-CSE','Electronics','relative',['MTH201']).
course('Digital Image Processing','Non-CSE','Electronics','relative',['MTH100','MTH201']).


pre_req_satisfy([],L) :-
        is_list(L).
pre_req_satisfy(L,[]) :-
        (not(is_list(L)) -> false;
                L==[]).
pre_req_satisfy(L1,L2) :-
        % L1 is the pre-requisites of the given course
        % L2 is all the courses done by the student previously.
        % This function tells us if the student can take the given course on basis of
the
        % pre-requisites(L1) of given course and the courses the student has done
yet (L2).

        [H|T] = L1,
        member(H,L2),
        pre_req_satisfy(T,L2).


read_course_list(A):-
        % This function is used to read a list of courses the student has done yet
as A.
        % This function can handle error also in case the student does not pass a
list as input.

        read(L),nl,
        (not(is_list(L)) ->
                (write('Enter list of courses you have done!'),nl,read_course_list(Temp),A
= Temp);
                A = L).
```

```
course_choice(S) :-
        % Here S is the Area of interest of a given course.
        % This function tells us if the student prefers this Area of interest (S).
        % If he prefers this Area of interest, then we add the fact interest(S).
        write('Do you prefer '),write(S),write(' courses ?'),nl,
        write('1. enter y for Yes'),nl,
        write('2. enter n for No'),nl,
        read(A),nl,
        (A=='y' ->
                assertz(interest(S));
                true).


display_recommended_courses(A):-
        % This function is used to print all the courses that are best suitable for the
student based on his preferences taken.
        % The ranking of the courses are done based on --> department > interest >
grading > pre_req_satisfy.

        forall((course(Name,Dept,Area,Grading,Pre_reqs), department(Dept),
interest(Area), grading(Grading), pre_req_satisfy(Pre_reqs,A)), (write(Name),nl)).


main :-
        % This is the main function.
        % Here, we first take inputs from the student to make a note of his
preferences in the form of facts.
        % Then we process those facts and finally displays a set of recommended
courses suitable for the student.
        retractall(department(_)),
        retractall(interest(_)),
        retractall(grading(_)),
        retractall(pre_reqs(_)),
        write('This is an Elective course recommendation system'),nl,
        write('Please answer the following questions below to let us know your
preferences'),nl,


        write('What is your name?'),nl,
        read(StudName),nl,
        write('Hi '),write(StudName),nl,
```

```
    % Here we ask the student does he prefer CSE courses or Non-CSE ones. If
he prefers CSE courses, then
    % we add the fact department('CSE'). Else we add the fact
department('CSE').
    write('Do you prefer CSE courses ?'),nl,
    write('1. enter y for Yes'),nl,
    write('2. enter n for No'),nl,
    read(A1),nl,
    (A1=='y' ->
        assertz(department('CSE'));
        assertz(department('Non-CSE'))),

    % Here we ask the student which are his area of interests (Machine
Learning, Databases, etc). If he says
    % yes in an Area of interest, then we add that fact using assertz statement.
    write('Select your area of interests :-'),nl,
    (course_choice('Machine Learning'),
    course_choice('Networking'),
    course_choice('Databases'),
    course_choice('Algorithms'),
    course_choice('Biology'),
    course_choice('Design'),
    course_choice('Social Science'),
    course_choice('Electronics')),

    % Here we ask the student if he prefers relative grading or absolute.
    % If he prefers relative grading, then we add the fact grading('relative').
    % Else we add the fact grading('absolute').
    write('Do you prefer relative grading or absolute grading courses ?'),nl,
    write('1. enter y for relative grading'),nl,
    write('2. enter n for absolute grading'),nl,
    read(A3),nl,
    (A3=='y' ->
        assertz(grading('relative'));
        assertz(grading('absolute'))),

    % Here we ask the student what are the courses he has done before. For
those
    % courses, we take the input as a list of course code (eg:-
['CSE101','ECE111']).
    % the list is stored in the variable A4.
    write('Enter the course-codes of the courses you have done as list'),nl,
    read_course_list(A4),nl,
```

```
    % Here, we run the recommendation function
display_recommended_courses/1. This function
    % simply prints all the courses that the student can take which are
matching with the preferences
    % the student has selected before.
    write("Below are the recommended courses for you according to your given
preferences :-"),nl,nl,
    display_recommended_courses(A4).
```

Explanation of code:-

First, we have stored the courses using the fact course(). The course fact contains --> Name of course, Course department, Area of interest of course, Absolute/Relative grading, pre_reqs.
The program runs by entering main (which is the main method). Tere, we first take inputs from the student to make a note of his preferences in the form of facts.
Then we process those facts and finally display a set of recommended courses suitable for the student.

First, we ask the student if he prefers CSE courses or Non-CSE ones. If he prefers CSE courses, then we add the fact department('CSE'). Else we add the fact department('CSE').

Then we ask the student which are his area of interests (Machine Learning, Databases, etc). If he says yes in an Area of interest, then we add that fact using assertz statement (eg :- If he says yes for Databases, then we add the fact interest('Databases')).

Then we ask the student if he prefers relative grading or absolute. If he prefers relative grading, then we add the fact grading('relative'). Else we add the fact grading('absolute').

Then we ask the student what are the courses he has done before. For those courses, we take the input as a list of course code (eg:- ['CSE101','ECE111']). The list is stored in variable A4.

Finally, we run the function which recommends/prints the courses that are preferable for the student using the rule/function display_recommended_courses(A).

Working Screenshots:-

```
C:\Users\Rohit\Desktop\Prolog_Assignments\Assignment_1>swipl code.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- main.
This is an Elective course recommendation system
Please answer the following questions below to let us know your preferences
What is your name?
|: rohit.

Hi rohit
Do you prefer CSE courses ?
1. enter y for Yes
2. enter n for No
|: y.
```

```
Select your area of interests :-
Do you prefer Machine Learning courses ?
1. enter y for Yes
2. enter n for No
|: y.

Do you prefer Networking courses ?
1. enter y for Yes
2. enter n for No
|: n.

Do you prefer Databases courses ?
1. enter y for Yes
2. enter n for No
|: y.

Do you prefer Algorithms courses ?
1. enter y for Yes
2. enter n for No
|: y.

Do you prefer Biology courses ?
1. enter y for Yes
2. enter n for No
|: n.
```

```
Do you prefer Design courses ?
1. enter y for Yes
2. enter n for No
|: n.

Do you prefer Social Science courses ?
1. enter y for Yes
2. enter n for No
|: n.

Do you prefer Electronics courses ?
1. enter y for Yes
2. enter n for No
|: n.

Do you prefer relative grading or absolute grading courses ?
1. enter y for relative grading
2. enter n for absolute grading
|: y.

Enter the course-codes of the courses you have done as list
|: ['CSE101','MTH100','ECE111','CSE102','MTH201','CSE121','CSE201','CSE231','CSE202','CSE222'].
```

```
Below are the recommended courses for you according to your given preferences :-

Machine Learning
Artificial Intelligence
Information Retrieval
DBMS
Algorithm Design and Analysis
Introduction to Programming
Approximation Algorithms
true.
```