

# CSE 643: Assignment 1

Abhimanyu Gupta - 2019226

## **Brief about working:**

The prolog program takes a number of inputs based on user choices and returns back a number of electives suitable for the user.

## **Steps to run the program:**

1. Load the program using consult or [code] in prolog terminal.
2. Run the advisory system using adviseElectiveCourses command.
3. Enter the department you are looking for from cse/mth.
4. Next, you would be asked a series of questions to determine your field of interest.
5. Quickly after it you would have to answer about electives you already did (using y/n).
6. At last, the program would return a list of electives suitable for you.

## **Working Examples:**

1)

```
?- adviseElectiveCourses.  
Welcome to Course Advisory System  
Follow the directions to find the right elective for you  
Select the department for which you want the course (cse/mth)  
|: mth.  
Are you interested in Pure Mathematics (y/n)  
|: y.  
Are you interested in Optimization (y/n)  
|: y.  
Are you interested in Probability and Statistics (y/n)  
|: n.  
Have you completed Real Analysis 1 course (y/n)?  
|: y.  
Have you completed Real Analysis 2 course (y/n)?  
|: n.  
Have you completed Abstract Alegbra 1 course (y/n)?  
|: y.  
Have you completed Abstract Alegbra 2 course (y/n)?  
|: n.  
Have you completed Discrete Mathematics course (y/n)?  
|: n.  
Have you completed Linear Optimisation course (y/n)?  
|: y.  
Have you completed Multivariate Calculus course (y/n)?  
|: n.  
Have you completed Graph Theory course (y/n)?  
|: y.  
Suitable Elective(s) for you in mth department:  
Real Analysis 2  
Abstract Alegbra 2  
Discrete Mathematics  
Multivariate Calculus  
true .
```

2)

```
?- adviseElectiveCourses.  
Welcome to Course Advisory System  
Follow the directions to find the right elective for you  
Select the department for which you want the course (cse/mth)  
|: cse.  
Are you interested in Security (y/n)  
|: y.  
Are you interested in Artificial Intelligence (y/n)  
|: y.  
Are you interested in Algorithms (y/n)  
|: n.  
Have you completed Computer Networks course (y/n)?  
|: y.  
Have you completed Foundation of Computer Security course (y/n)?  
|: n.  
Have you completed Network Security course (y/n)?  
|: n.  
Have you completed Artificial Intelligence course (y/n)?  
|: y.  
Have you completed Machine Learning course (y/n)?  
|: y.  
Have you completed Deep Learning course (y/n)?  
|: n.  
Have you completed Statistical Machine Learning course (y/n)?  
|: n.  
Suitable Elective(s) for you in cse department:  
Foundation of Computer Security  
Network Security  
Deep Learning  
Statistical Machine Learning  
true .
```

**Code snippets:**

```

1 % Abhinanyu Gupta - 2019226
2 % CSE643 - Artificial Intelligence - Assignment 1
3
4 adviseElectiveCourses :-
5     clear_all,
6
7     write('Welcome to Course Advisory System\n'),
8     write('Follow the directions to find the right elective for you\n'),
9
10    write('Select the department for which you want the course (cse/mth)\n'),
11
12    read(Department),
13    (
14        (Department = 'cse' -> check_cse_domains_interest);
15        (Department = 'mth' -> check_mth_domains_interest);
16        write('Invalid department!!, program exiting!\n'),fail
17    ),
18
19    forall(interested(X), check_available_electives(X)),
20
21    recommend_courses(Courses),
22    (
23        is_empty(Courses) -> write('Sorry, nothing to recommend at the moment\n');
24        format('Suitable Elective(s) for you in ~w department:\n', [Department]), show_list(Courses)
25    ),
26
27    clear_all.
28
29 :- dynamic interested/1, course_done/1, course_not_done/1, is_empty/1, recommend_course/1.
30
31 clear_all :-
32     retractall(interested(_)),
33     retractall(course_done(_)),
34     retractall(course_not_done(_)),
35     retractall(recommend_course(_)).
36
37 recommend_courses([Head | Tail]) :-
38     retract(recommend_course(Head)), recommend_courses(Tail).
39 recommend_courses([]).
40
41 show_list([Head | Tail]) :-
42     format('~w~n', [Head]), show_list(Tail).
43 show_list([]).
44
45 isempty([]).
46
47
48

```

```

50 check_course(Q) :-
51     (
52         course_done(Q) -> true;
53         (
54             course_not_done(Q) -> fail;
55             ask_question_to_check(Q)
56         )
57     ).
58
59 ask_question_to_check(Q) :-
60     format('Have you completed ~w course (y/n)?\n', [Q]),
61     read(A),
62     (
63         (A = 'y') -> assert(course_done(Q));
64         assert(course_not_done(Q))
65     ).
66
67
68 check_cse_domains_interest :-
69     write('Are you interested in Security (y/n)\n'),
70     read(S),
71     (
72         (S = 'y') -> assert(interested(s));
73         !
74     ),
75
76     write('Are you interested in Artificial Intelligence (y/n)\n'),
77     read(AI),
78     (
79         (AI = 'y') -> assert(interested(ai));
80         !
81     ),
82
83     write('Are you interested in Algorithms (y/n)\n'),
84     read(A),
85     (
86         (A = 'y') -> assert(interested(a));
87         !
88     ).
89
90 check_mth_domains_interest :-
91     write('Are you interested in Pure Mathematics (y/n)\n'),
92     read(PM),
93     (
94         (PM = 'y') -> assert(interested(pm));
95         !
96     ),
97
98     write('Are you interested in Optimization (y/n)\n'),
99     read(O),
100     (
101         (O = 'y') -> assert(interested(o));
102         !
103     ),
104

```

```

185     write("Are you interested in Probability and Statistics (y/n)\n"),
186     read(P5),
187     (
188         (P5 = 'y') -> assert(interested(ps));
189         !
190     ).
191
192 check_available_electives(s) :-
193     check_course('Computer Networks'),
194     (
195         course_not_done('Computer Networks') -> assert(recommend_course('Computer Networks'));
196         (
197             check_course('Foundation of Computer Security'),
198             (
199                 course_not_done('Foundation of Computer Security') -> assert(recommend_course('Foundation of Computer Security'));
200                 !
201             )
202         )
203     ),
204
205     check_course('Computer Networks'),
206     (
207         course_not_done('Computer Networks') -> assert(recommend_course('Computer Networks'));
208         (
209             check_course('Network Security'),
210             (
211                 course_not_done('Network Security') -> assert(recommend_course('Network Security'));
212                 !
213             )
214         )
215     ),
216
217     check_course('Foundation of Computer Security'),
218     (
219         course_not_done('Foundation of Computer Security') -> assert(recommend_course('Foundation of Computer Security'));
220         (
221             check_course('Security Engineering'),
222             (
223                 course_not_done('Security Engineering') -> assert(recommend_course('Security Engineering'));
224                 !
225             )
226         )
227     ).
228
229 check_available_electives(ai) :-
230     check_course('Artificial Intelligence'),
231     (
232         course_not_done('Artificial Intelligence') -> assert(recommend_course('Artificial Intelligence'));
233         !
234     ),
235
236     check_course('Machine Learning')

```

```

157
158     check_course('Machine Learning'),
159     (
160         course_not_done('Machine Learning') -> assert(recommend_course('Machine Learning'));
161         (
162             check_course('Deep Learning'),
163             (
164                 course_not_done('Deep Learning') -> assert(recommend_course('Deep Learning'));
165                 !
166             )
167         )
168     ),
169
170     check_course('Statistical Machine Learning'),
171     (
172         course_not_done('Statistical Machine Learning') -> assert(recommend_course('Statistical Machine Learning'));
173         !
174     ).
175
176
177 check_available_electives(a) :-
178
179     check_course('Algorithm Design and Analysis'),
180     (
181         course_not_done('Algorithm Design and Analysis') -> assert(recommend_course('Algorithm Design and Analysis'));
182         (
183             check_course('Modern Algorithm and Design'),
184             (
185                 course_not_done('Modern Algorithm and Design') -> assert(recommend_course('Modern Algorithm and Design'));
186                 !
187             )
188         )
189     ),
190
191     check_course('Algorithm Design and Analysis'),
192     (
193         course_not_done('Algorithm Design and Analysis') -> assert(recommend_course('Algorithm Design and Analysis'));
194         (
195             check_course('Randomised Algorithm'),
196             (
197                 course_not_done('Randomised Algorithm') -> assert(recommend_course('Randomised Algorithm'));
198                 !
199             )
200         )
201     ),
202
203     check_course('Algorithm Design and Analysis'),
204     (
205         course_not_done('Algorithm Design and Analysis') -> assert(recommend_course('Algorithm Design and Analysis'));
206         (
207             check_course('Introduction to Graduate Algorithm'),
208             (
209                 course_not_done('Introduction to Graduate Algorithm') -> assert(recommend_course('Introduction to Graduate Algorithm'));

```

```

282
283     check_course('Algorithm Design and Analysis'),
284     (
285         course_not_done('Algorithm Design and Analysis') -> assert(recommend_course('Algorithm Design and Analysis'));
286         (
287             check_course('Introduction to Graduate Algorithm'),
288             (
289                 course_not_done('Introduction to Graduate Algorithm') -> assert(recommend_course('Introduction to Graduate Algorithm'));
290                 !
291             )
292         )
293     ).
294
295
296 check_available_electives(pm) :-
297
298     check_course('Real Analysis 1'),
299     (
300         course_not_done('Real Analysis 1') -> assert(recommend_course('Real Analysis 1'));
301         (
302             check_course('Real Analysis 2'),
303             (
304                 course_not_done('Real Analysis 2') -> assert(recommend_course('Real Analysis 2'));
305                 !
306             )
307         )
308     ),
309
310     check_course('Abstract Alegbra 1'),
311     (
312         course_not_done('Abstract Alegbra 1') -> assert(recommend_course('Abstract Alegbra 1'));
313         (
314             check_course('Abstract Alegbra 2'),
315             (
316                 course_not_done('Abstract Alegbra 2') -> assert(recommend_course('Abstract Alegbra 2'));
317                 !
318             )
319         )
320     ),
321
322     check_course('Discrete Mathematics'),
323     (
324         course_not_done('Discrete Mathematics') -> assert(recommend_course('Discrete Mathematics'));
325         (
326             check_course('Graph Theory'),
327             (
328                 course_not_done('Graph Theory') -> assert(recommend_course('Graph Theory'));
329                 !
330             )
331         )
332     ).
333
334 check_available_electives(o) :-
335

```

```

258     course_not_done('Linear Optimisation') -> assert(recommend_course('Linear Optimisation'));
259     !
260 },
261
262 check_course('Multivariate Calculus'),
263 (
264     course_not_done('Multivariate Calculus') -> assert(recommend_course('Multivariate Calculus'));
265     (
266         check_course('Convex Optimisation'),
267         (
268             course_not_done('Convex Optimisation') -> assert(recommend_course('Convex Optimisation'));
269             !
270         )
271     )
272 ),
273
274 check_course('Graph Theory'),
275 (
276     course_not_done('Graph Theory') -> assert(recommend_course('Graph Theory'));
277     !
278 ).
279
280 check_available_electives(ps) :-
281
282     check_course('Probability & Statistics'),
283     (
284         course_not_done('Probability & Statistics') -> assert(recommend_course('Probability & Statistics'));
285         (
286             check_course('Stochastic Processes and Applications'),
287             (
288                 course_not_done('Stochastic Processes and Applications') -> assert(recommend_course('Stochastic Processes and Applications'));
289                 !
290             )
291         )
292     ),
293
294     check_course('Probability & Statistics'),
295     (
296         course_not_done('Probability & Statistics') -> assert(recommend_course('Probability & Statistics'));
297         (
298             check_course('Statistical Inference'),
299             (
300                 course_not_done('Statistical Inference') -> assert(recommend_course('Statistical Inference'));
301                 !
302             )
303         )
304     ).
305
306 check_available_electives(X) :-
307     true.
308
309
310

```