

Concepts of Programming Languages, Spring 2012  
Prolog Project

*Submission: 20.4.2012 (11:59 pm)*

The project will put your knowledge in Prolog to the test. Before proceeding, make sure that you read each section carefully.

## Instructions

Please read and follow the instructions carefully:

- a) The project should be implemented using Prolog (SWI-Prolog) based on the syntax discussed in class and described in the online lecture notes.
- b) This is a group project. You can work in groups of **maximum 3** members. You have to register your group through <http://goo.gl/1EwHp>. Non-registered students will be assigned to groups randomly.
- c) You have to write a **formal report** on your project covering the following points:
  - a brief formal description of the *main* predicates used in your implementation.
  - a listing of the *helper* predicates that you called from your main predicates. You are allowed to use built-in Prolog predicates, like `member/2` without explanation.
- d) Every team has to submit a soft copy of the following files (in a zipped directory) through the submission system on the MET website:
  - your Prolog project **source file** named after your team code, e.g. `T10.pl`
  - the report file (.doc or .pdf) also named after your team code.
- e) Every team has to submit a **proper hard copy** of the following:
  - the report
  - a printout of the source file
- f) You are always welcome to discuss the project with the TAs. You must work with your team members only. Do not exchange information with other teams or individuals.
- g) Cheating and plagiarism will be strictly punished with a grade of 0 in the project.
- h) Once you submit the project, you will be appointed a date to show up with your team members for an **oral evaluation** of your project. The evaluation will cover practical and technical details as well as theoretical concepts concerning Prolog and the general features of the logic programming paradigm.
- i) Please respect the **submission deadline** marked at the beginning of the document as well as the date of the oral evaluation set by your TA. Any delay will result in a rejection of the submission and a cancellation of the oral evaluation.

## Project Description

Your task is to develop a scheduling system for the GUC. The system should find an appropriate schedule for the courses taking into consideration these restrictions.

- Each group should have at least one day-off.
- No group should have to stay a full day on campus, **for example** if the schedule starts at the first slot, then it should end by-maximum after the fourth slot.
- If a course has a lab and a tutorial then the tutorial should precede the lab.
- If a course has 2 lectures, they should be scheduled on different days.
- The tutorials/labs of a course should be scheduled on a different day than the lecture.

On the course material page, you will find the knowledge base containing different information about the courses being studied in the GUC by the MET fourth semester group. In the file `course_catalogue.pl`, the main predicates used are:

- `course(C,G)` denotes that the course `C` is taught to group `G`.
- `group(Group,Tutorial_Groups)` denotes the fact that the group `Group` has the list `Tutorial_Groups` of tutorial groups.
- `labs(Labs_list)` denotes the fact that the available labs are stored in the list `Labs_list`.
- `lecturehalls(Lecturehalls_List)` denotes the fact that the available lecture halls are stored in the list `Lecturehalls_List`.
- `tutorialrooms(Tutorialrooms_List)` denotes the fact that the available tutorial rooms are stored in the list `Tutorialrooms_List`.
- `course_meetings(Course,Type,Tutorial_Group,Instructor,Description)` denotes that the course `Course` has the meeting of the type `Type` which could be a lecture, tutorial or lab. This meeting is scheduled for the tutorial group `Tutorial_Group` and is taught by the instructor `Instructor`. `Description` provides more info about the meeting.
- `totalcourses(Total_number_of_courses)`, `totalmeetings(Course,Meetings)`, `totalgroups(Total_number_of_groups)` denote the total number of courses, meetings in a course and groups respectively.

**The basic prolog predicate is solve:** which should formulate and print all the possible schedules. The schedule should be printed slot-wise. For each slot the list of the courses, their respective location, group and instructor should be printed too.

### Example:

?-solve.

saturday 1: comm401,h3,all\_group\_4,dr\_amr\_talaat,lecture

saturday 2: csen401,c7201,t07,nermeen\_abdel\_aziz,lab, csen402,c6301,t08,nehal,tutorial, maths4,c2201,t09,khaled\_mohamed,tutorial

saturday 3: csen401,c7201,t08,injy\_khairi,lab, csen402,c6301,t07,nehal,tutorial, maths4,c2201,t10,hany\_elsharkawy,tutorial

saturday 4: csen401,c7201,t09,nermeen\_abdel\_aziz,lab, csen402,c6301,t10,noura\_maghawry,tutorial, elct401,c2201,t08,yasmmine\_hassan,tutorial,

saturday 5:

sunday 1: comm401,c6301,t07,mohamed\_esameldin,tutorial, csen401,c7201,t10,injy\_khairi,lab

sunday 2: comm401,c6301,t08,sally\_mahmoud,tutorial, csen403,c7201,t07,nada\_sharaf,lab

sunday 3: comm401,c6301,t09,mohamed\_esameldin,tutorial, csen403,c7201,t08,nada\_sharaf,lab, maths4,c2201,t07,khaled\_mohamed,tutorial

sunday 4: comm401,c6301,t10,sally\_mahmoud,tutorial, csen403,c7201,t09,nada\_sharaf,lab, maths4,c2201,t08,tarek\_mounir,tutorial

sunday 5: csen402,c6301,t09,noura\_maghawry,tutorial, elct401,c7201,t08,samar\_mohamed,lab

monday 1: csen401,h3,all\_group\_4,prof\_slim,lecture

monday 2: csen402,h3,all\_group\_4,dr\_cherif\_salama,lecture

monday 3: csen403,c7201,t10,nada\_sharaf,lab elct401,c6301,t07,yasmmine\_hassan,tutorial

monday 4: elct401,c7201,t07,ahmed\_ragheb,lab elct401,c6301,t09,sara\_mohamed,tutorial

monday 5:

tuesday 1: csen403,h3,all\_group\_4,prof\_slim,lecture

tuesday 2: elct401,c7201,t09,ahmed\_ragheb,lab, elct401,c6301,t10,sara\_mohamed,tutorial

tuesday 3: elct401,c7201,t10,samar\_mohamed,lab

tuesday 4: maths4,h3,all\_group\_4,prof\_bauman,lecture

tuesday 5:

wednesday 1: elct401,h3,all\_group\_4,dr\_yasser\_hegazy,lecture

wednesday 2: maths4,h3,all\_group\_4,prof\_bauman,lecture

wednesday 3:

wednesday 4:

wednesday 5:

thursday 1:

thursday 2:

thursday 3:

thursday 4:

thursday 5:

## Grading and Evaluation

The evaluation of the practical part of the project, i.e the implementation, is the same for the whole team. However, the oral evaluation is team-member-specific. Your oral evaluation will affect your overall project grade. So be prepared :)

## Tips

Here is a list of the do's and don't for a better performance in the project as well as in the course:

- Do not think in a procedural way, i.e. do not translate your Java ideas into Prolog code.
- Remember that Prolog is a logic programming language. Sometimes it will be useful to express your rule in natural language first. Maybe then you will see the Prolog code :)
- Work personally on the code. The more you are involved in the coding process, the more you become aware of the advantages and drawbacks of Prolog and the logic programming paradigm as a whole.
- Read carefully the lecture notes and revise the slides. They include a lot of useful information for you, not only concerning technical details but also practical and technical issues.
- Start working early on the project. Do not get carried away and keep track of time. It would be handy if you planned ahead your project and worked out a reasonable timeline.
- Divide the work among your team members. Coordinate your efforts together and meet regularly to update each other with the progress.

Finally, we wish you the best of luck!