Term Project for CMPE 532

Due Date: December 30th 2024, 11:00 a.m. Projects will be checked at a later date.

Timetable maker for students, minimizing clashes

Implement a solution in Eclipse, using its constraint solving capability to solve the clash minimization problem. We need to assign a student to course groups such that the number of clashes are minimized.

Information about courses and sections (groups) should be given as **course(coursecode, group_no, day, period).** So, for example, the course CMPE532 meets on Tuesdays at 10:30, 11:30 and 12:30 (periods 3, 4 and 5). We would represent this data by using three facts in the Eclipse database:

```
course(cmpe532, 1, tuesday, 3).
course(cmpe532, 1, tuesday, 4).
course(cmpe532, 1, tuesday, 5).
```

Actual data for Fall 2024 is provided separately. Use this as the database of courses. The courses for which data is provided are: CMPE107, CMPE112, CMPE211, CMPE231, CMPE353, CMPE471.

Define a predicate **assign(List_of_courses, No_of_clashes, Schedule, Clashes)** where:

- **List_of_courses** is the input
- **No_of_clashes** the ***smallest*** achievable number of clashes (output)
- **Schedule** the list of **(Course,Group_no)** pairs that result in **No_of_clashes** clashes
- **Clashes** is the **(day,period)** list where there is a clash

For example, the query

?- assign([cmpe107,cmpe471,cmpe112], N, S,C).

Should return *something like*

N=2

S=[(cmpe107,1),(cmpe471,2),(cmpe112,1)]

C=[(monday,3), (monday,4)]

What to hand in: A report that contains the problem definition, description of your program (each predicate that you wrote and what it does), as well as sample runs for 2,3,4,5 and 6 courses, and your program as an appendix.

Bonus (5 points OVERALL): Print the best timetable (the one with the least number of clashes) on the screen as a table, showing the days of the week, periods, and courses.