UVEC Summer 2018

Programming

Homework Deadline Scheduling

Introduction

Each year students in software engineering at the University of Victoria are burdened with the consequences of an uncoordinated department which causes many deadlines to land at once. This year, the department head has decided that there has to be a better way and she has commissioned software which will generate deadlines for professors in each class by taking in data from course outlines.

Problem Statement

You will design a user interface to display the resulting schedule, as well as the program which generates it. **There are no restrictions on frameworks or languages.** From the course syllabi for the 6 classes in the 2A semester of software engineering data has been extracted in the following form:

```
<DEPARTMENT><INT> L<INT>-<INT> A<INT>-<INT> M<INT>-<INT> F<INT>
PA<INT>
```

Which is the course code as an abbreviation of the department paired with a 3 digit integer (ie SENG265) followed by the number and cumulative weight of the labs, assignments, midterms, final, and the previous average of the class.

You will be able to use any input you choose, although a simple text file is appropriate for this problem. You may paste the following data into it:

```
ELEC260 L0-0 A5-10 M2-30 F60 PA68
SENG265 L0-0 A4-40 M3-60 F0 PA75
CHEM101 L4-40 A10-10 M1-20 F30 PA72
STAT260 L0-0 A5-10 M3-45 F45 PA60
MATH122 L0-0 A5-15 M2-30 F55 PA69
CENG255 L4-20 A0-0 M2-30 F50 PA80
```

You must be able to demonstrate that the schedule changes with different data!

As a bonus you may schedule final exams as a function of previous averages in the class (ie classes with lower averages have more time leading up to them)

The scheduling algorithm must meet the following requirements:

- No deadlines earlier than 10th September or later than 30th November
- No deadlines during reading break 12th-14th November
- No assignments for the same class due on the same week
- No labs in consecutive weeks
- No assignments due the night before a midterm day
- No Final exams outside 3-17th December
- No Finals within 1 day of each other

Deliverables

A demo of your code, and a presentation which justifies your design of the interface as well as the logical construction of your program.

Scoring

- Presentation **40%**
    - Clear communication
    - Explanations do not assume previous knowledge
    - Demonstrate consideration of the user and future of the project
- Code Functionality **60%**
    - Meets requirements
        - Changes output according to data input
- Bonus:
    - Up to **10% bonus** for final exam scheduling or other features