CSE 241 Homework Assignment 3

DUE

April 17, 2019, 23:55

Description

• This is an individual assignment. Please do not collaborate.
• If you think that this document does not clearly describe the assignment, ask questions before it's too late.
You won't be given a chance to correct any mistakes.
Scope: University Management System
You are expected to develop university management system.
Main Problem: Lecture Arrangement
Problems:

1. Each lecturer gives 3 lessons.
2. There are 2 type courses; Mandatory and Elective.
3. Each course belongs to a field. Lecturers have professions. It can be more than one.
4. Mandatory courses are assigned to lecturers according to matching field of course and profession of lecturer.
5. Lecturers have ability to propose new courses in the type of elective course.
6. Administrator arranges which course is processed in which classroom.
7. Administrator arranges time table of courses.

Models:

1. **Lecturer:** Keeps lecturer information and performs functionality of lecturer. (Class)
   Information: Name, surname, personal_id, title, proffesions, courses
   Functionality:
   - proposeCourse(): Lecturer proposes new courses. If courses given by the lecturer is less than 3, proposed course(s) is added to overall courses list.
   - assignCourse(): Course is assigned to lecturer. If Lecturer gives 3 courses, assignment is not allowed. If lecturer gives less than 3 courses and s/he has not proposed any elective courses, program asks her/him to propose an elective course.
2. **Administrator:** Arranges lecture, lecturer and classroom.
   Information: pid, password
   Functionality:
   - arrangeClassroom(): determine which course is processed which classrooms. It pays regard to timetable of courses.
   - arrangeTimeTable(): determine lecture dates of each courses.
3. **Course:** keeps course information. (Struct)
   Information: id, name, code, credit, total hours, lecture dates, field, isMandatory
4. **Classroom:** keeps classroom information. (Struct)
   Information: id, c_no, capacity, student_inroom

How to Test

1- load TXT files.
2- Objects are initialized.
3- Each lecturer will propose elective courses:
(input)>-propose pid DeepLearning 521 3 3 AI
Mean: {name: DeepLearning, code: 521, credit: 3, totalHours: 3, field: AI, pid: personal id of lecturer}
4- Assign courses
(input)>-assign 312 4
Mean: {lid: 312, course_id: 4} -> 3 possible results => 1- Error: no lecture or course, 2- Block: field and profession mismatch, 3- DONE
(input)>-assign
Mean: automatically match lecturers and courses in restriction of field and count of given course
5- Arrange timetable
(input)>-timetable 4
Mean: randomly assign dates to the course -> 3 possible results => 1- Error: no course, 2- Assigned Before: (4)Math Wed_15-17,Fri_12-14, 3- DONE: (4)Math Wed_15-17,Fri_12-14
(input)>-timetable

Mean: Automatically arrange timetable for all courses and then show all of them. If arranged before, then show all of them.

6- arrange classroom

(input)>-arrangeC 4 301

Mean: {course_id:4, class_id:301} -> 3 possible results => 1- Error: no class or no course or timetable not arranged yet, 2- Block: class is not available, there is another class assigned, 3- DONE: (4)Math in 301 at Wed_15-17,Fri_12-14

(input)>-arrangeC 4 301,302

Mean: {course_id:4, class_id:301,302} -> 3 possible results => 1- Error: no class or no course or timetable not arranged yet, 2- Block: class is not available, there is another class assigned for this date, 3- DONE: (4)Math in 301 at Wed_15-17, in 302 at Fri_12-14

(input)>-arrangeC

Mean: Automatically arrange all courses and classes then show all of them. If arranged before, then show all of them.

Remarks

- Do not use any elements which is not covered in class.
- Do not submit your code without testing it with several different scenarios.
- Write comments in your code.
- Prepare a Makefile for your programme.

Turn in:

- Source code of a complete C++ program. Name of the file should be in this format: <full_name>_<id>.cpp.
- Example: gokhan_kaya_000000.cpp. Please do not use any Turkish special characters.
- You don't need to use an IDE for this assignment. Your code will be compiled and run in a command window.
- Your code will be compiled and tested on a Linux machine(Ubuntu). GCC will be used
- Make sure you don't get compile errors when you issue this command : g++ <full_name>_<id>.cpp.
- A script will be used in order to check the correctness of your results. So, be careful not to violate the expected output format.
- Provide comments unless you are not interested in partial credit. (If I cannot easily understand your design, you may loose points.)
- You may not get full credit if your implementation contradicts with the statements in this document.

Late Submission

- (0,24] hours: -20%
- (24,48] hours: -40%
- (48,72] hours: -60%
- (72,-) hours: -100%

Grading (Tentative)

- Max Grade : 100.
- Multiple tests(at least 5) will be performed.

All of the followings are possible deductions from Max Grade.

- #define HARD_CODED_VALUES -10.
- No submission: -100. (be consistent in doing this and your overall grade will converge to N/A) (To be specific: if you miss 3 assignments you'll get N/A)
- Compile errors: -100.
- Irrelevant code: -100.
- Major parts are missing: -100.
- Unnecessarily long code: -30.
- Using language elements and libraries which are not allowed: -100.
- Not caring about the structure and efficiency: -30. (avoid using hard-coded values, avoid hard-to-follow expressions, avoid code repetition, avoid unnecessary loops).

• Significant number of compiler warnings: -10.
• Not commented enough: -5. (Comments are in English).
• Source code encoding is not UTF-8 and characters are not properly displayed: -5. (You can use ‘Visual Studio Code’, ‘Sublime Text’, ‘Atom’ etc. . . Check the character encoding of your text editor and set it to UTF-8).
• Missing or wrong output values: Fails the test.
• Output format is wrong: -30.
• Infinite loop: Fails the test.
• Segmentation fault: Fails the test.
• Fails 5 or more random tests: -100.
• Fails the test: deduction up to 20.
• Prints anything extra: -30.
• Unwanted chars and spaces in output.txt: -30.
• Submission includes files other than the expected: -10.
• Submission does not follow the file naming convention: -10.
• Sharing or inheriting code: -200