CENG 206 - Spring 2020
Project-2

(to be implemented in **C++,** Due 17/05/2020 @23:55)

As you recall, at the beginning of the semester we have explained that one of the goals of CENG206 Programming Languages course is to increase your ability to learn new languages. In this project, you will learn a new language, C++, and implement a small project. You will find some supplementary materials, such as a tutorial and Lab recitation, about C++ in Aybuzem. You can also get benefit from any other source on the Web.

You are going to work with the same project group as in Project-1. However, if your teammate does not collaborate with you properly or take advantage of you please report me by e-mail, these mails will remain hidden. We can apply peer-grading according to your feedback. Your feedback is important. Therefore the student who did not do anything in the project gets 0 depending on your feedback. Fairness in class depends on your responsible behavior. It is our and your responsibility to maintain fairness in the class.

In this project, you will implement an automatic course planner for a curriculum semester of a department. In the curriculum, there are several courses for each year of the curriculum. Your program will assign a classroom and a time slot for each course in the curriculum. Courses in the same year should not be intersected with each other (i.e., see CENG Spring curriculum). There can be some intersection between courses of different years. There are 2 different types of the classroom; big and small. Mandatory courses in the curriculum should be assigned to a big classroom. Elective courses can be assigned to a big or small classroom according to the availability of them. There is a limited number of dedicated classrooms for the department. The number of each type of classroom should be read from a file. Besides, for each weekday there are 2-time slots available; morning and afternoon. So there are in total 10 time slots available to place a course (5 weekdays*2 time slot).

In the department curriculum, there are some service courses which are given by another department at the university. The time slot of these courses is fixed and predefined. Therefore, you cannot assign different time slots for those courses other than the requested time slot. Furthermore, some instructors may not available for some time slots. Thus, your program should respect these busy time slots for the respective courses. All of these constraints should be taken from a file. You should not assume anything in prior and not use any hard-coded parameter / value in your code.

In the end, your program will produce a file that contains a course schedule for the department. In this schedule, there should not be any intersection between courses for a year of the curriculum and respect to all constraints. If your program cannot find any possible schedule it will print a message "There is no way to make a perfect schedule for the department." **To fit the course schedule you may increase the number of classrooms via editing the file.**

GUI Part: 25 pts.
You will implement a GUI system that can take inputs directly from the user instead of reading them from a file. The design parameters are totally up to you (How it looks like, how many buttons, size of the window etc.). C++ does not have a library for creating a desktop application. However, by utilizing some library extensions you can design and implement a GUI for your C++ program. The followings are some tools/libraries that you can use for the GUI part of your project.

http://nanapro.org/en-us/

https://juce.com/learn/tutorials

https://www.wxwidgets.org/

https://www.qt.io/

Qt is one of the most popular libraries for creating GUI for C++ programs. It also provides an IDE, Qt Creator, in which you can easily design your GUI by drag&drop fashion and easily integrate your C++ program to it.

Notes: There will be 2 checkpoints.

1. In the first checkpoint (8/05/2020), your program should read related data from the files appropriately and create necessary objects of abstract data types. A small report which contains the pseudocode of your algorithm and your class diagrams and class interactions. **You should upload your report together with related sources codes in a zip file.**
2. In the second checkpoint (17/05/2020), your program should be completed. You must upload the final code with the report into Aybuzem. We will compile and test your programs. Only successfully compiled and working programs can take points.

Implementation details:

1. Your C++ program must include the following programming features;
    a. Exception handling; for file operations and any other necessary cases you should adopt exception handling mechanism properly.
    b. Default parameter; One of your functions in the program should support function with default arrangements.
    c. Operator overloading; you should use at least one operator overloaded function. For example, you can use it for checking some values of two objects of the same class by overloading == operator. However, you are free to implement which operator you want.
    d. Inline function; Implement one of your functions by using inline function.
    e. Constructor/destructor; For each class you implemented should include a constructor and destructor.
    f. Vectors; you are suggested to use "vector" data structure for array implementations in your code. Try to efficiently use vectors by using c++ functions, such as 'sort' if you need to sort the elements in the vector.

    *In your report, you should show the usage of these features with small code segments. For example, you can put some portion of code that depicts the usage of Exception Handling in your project. For each of the features, you must put the respective code segment in your report.

2. You should implement your program with proper usage of headers and source codes. You may have the following files (not necessarily) for the project: (you can add more class / header / source files according to the design of your algorithm.)
    a. course.hpp
    b. electiveCourse.hpp
    c. compulsoryCourse.hpp
    d. classroom.hpp
    e. bigClassroom.hpp
    f. smallClassroom.hpp
    g. scheduler.hpp
    h. scheduler.cpp
    i. main.cpp
3. Format of the files;
    a. Courses.csv contains all courses in the curriculum. Each line has 7 items separated by ';', from left to right; code of the course, name of the course, the year of the semester, credit, C: compulsory or E: Elective, D: department or S: service, name of the instructor.
    CENG104;COMPUTER PROGRAMMING II;1;6;C;D;OGR.GOR. YUSUF EVREN AYKAC
    CHEM101;GENERAL CHEMISTRY;1;5;C;S;DOC.DR. NURAY CELEBI
    CENG316;PARALLEL PROGRAMMING;3;5;E;D;DR. OGR. UYESI FAHREDDIN SUKRU TORUN
    …

b. Service.csv contains time slot of service courses. Format of it as follows:

```
CHEM101;Tuesday;Morning
MATH102;Monday;Afternoon
…
```

c. Busy.csv contains the busy time slots for the respective course. You cannot assign a course to the specified time slot. That is, for example, CENG104 should not be placed on Tuesday according to the below file.

```
CENG104;Tuesday;Morning
CENG104;Tuesday;Afternoon
CENG316;Friday;Afternoon
CENG206;Monday;Morning
```

d. Classroom.csv contains the number of classrooms for each type. For example, the following file means that there are 2 big classrooms and 1 small classroom.

```
bigClass;2
smallClass;1
```

4. If you want to develop your project without GUI, then your output should be ordered by weekdays and then classroom Id. The sample output:

```
Monday Morning bigClass1 CENG104
Monday Morning bigClass2 CENG202
Monday Morning smallClass1 CENG310
Monday Afternoon bigClass1 MATH102
Monday Afternoon bigClass2 ENGR202
Monday Afternoon smallClass1 CENG415
Tuesday Morning bigClass1 CHEM101
Tuesday Morning bigClass2 --------
Tuesday Morning smallClass1 CENG317
Tuesday Afternoon bigClass1 ENG254
Tuesday Afternoon bigClass2 CENG204
Tuesday Afternoon smallClass1 --------
Wednesday Morning bigClass1 TIT101
Wednesday Morning bigClass2 --------
…
```