

2021.2 Object Oriented Programming

Project #1 [Due: Oct. 12 (Tuesday) 11:59pm]

Submission.

1. If you are supposed to do project 1 as individual project

(see our class webpage for team project assignment):

- eClass submission - you should submit one zip file that contains two directories as follows
 - (i) directory "prob1" - this directory should contain a .doc or .hwp file which has class specification (i.e. header files) of IIKH system.
 - (ii) directory "prob2" - this directory should contain
 - source codes(.cpp, .h, and solution files(.sln, and etc)) in visual studio format.
If you use g++, you should include Makefile that can be used for compilation.
 - file1.txt you used
 - README.txt - very briefly explain how to compile and execute the source code.
- You should submit the above project results through eclass system before the deadline.

2. If you are supposed to do project 1 as team project:

- Only the team leader should submit the project results (including source code files and project report) through eclass system. A team member should give presentation. Please see and follow team project information and presentation schedule that will be available on eclass OOP course page.)
- You should use English for project presentation and project report.

Problems

Prob#1. This problem is to perform Object Oriented Design of a system. Write a C++ header file that contains class specification (without implementation details) for developing IIKH(Interactive Interlligent Kitchen Helper) that is described in Chapter 3 of Timothy Budd's book "Intro. to Object Oriented Programming". You need to just create classes that contain member variables and member functions in the header file. However, you do not need to implement function body of the member functions in this problem.

- In this problem, it is essential to choose a set of components (classes) and their member variables/functions.
- The names of classes, variables, and functions should reflect their meanings.
- You don't need to submit detailed implementation. However, you do have to insert appropriate comments in your class specifications.
- Above policies will be considered when grading your homework.

- The result of prob#1 should be just one C++ header file that contains one class or several classes. The class specification may contain "struct" types if necessary.

Prob#2. The problem#2 is to fully develop a Student Information Management System. This program should support inserting/searching/displaying(sorted) student information. After the program is terminated, the inserted student information should be maintained. This requires you to use file processing. Do not use external library for the file processing. Use only C++ built-in libraries. The resulting C++ source code you submit to eclass must be compilable and executable.

Command Line Execution

The executable file takes one file as argument.

```
> a.exe file1.txt
```

a.exe is an executable file. file1.txt contains the student information saved during the program execution. You are free to organize the format of file1.txt. If file1.txt does not exist when execution, your program should create file1.txt. If file1.txt exists, you should use it.

(i) Main Menu

Following menu should be displayed when you execute the above command line.

```
1. Insertion
2. Search
3. Sorting Option
4. Exit
> _
```

(ii) If "1" (Insertion) is selected in the main menu, take input as follows and store the information into file1.txt.

```
Name ?
Student ID (10 digits)?
Birth Year (4 digits) ?
```

Department ?

Tel ?

"Name" has up to 15 (English) characters , "Student ID" should be exactly 10 digits where first 4 digits represent admission year. "Birth Year" should be exactly 4 digits. "Tel" has up to 12 digits. After user's input is completed, the program should display the main menu and wait for user's input again. Name and Student ID should not be blank. If the same student id is provided, an error message "Error : Already inserted" should be printed.

(iii) If "2" (Search) is selected, display following search menu and take additional user input (keyboard).

- Search -

1. Search by name
2. Search by student ID (10 numbers)
3. Search by admission year (4 numbers)
4. Search by department name
5. List All

> 4

Department name keyword? Computer Engineering

Name	StudentID	Dept	Birth Year	Tel
Lisa Simpson	2006303001	Computer Engineering	2000	01012345678
Gildong Hong	2004303077	Computer Engineering	1999	01187651234

After getting a number as a user input in the above menu, your program should get additional string input for searching for student information stored in file1.txt and display the search result. If you select "**5. List All**", informations for all students are displayed. The display order for student records should be based on the selection of "sorting option" (The default is "Sort by Name". This means that initially the display order should be sorted by name when no sorting option was specified). Go back to main menu after displaying the student records.

(iv) If "3" (Sorting Option) is selected , display following menu and take user input.

```
- Sorting Option
1. Sort by Name
2. Sort by Student ID
3. Sort by Admission Year
4. Sort by Department name
>
```

Go back to the main menu, after selecting the sorting option..

You may set the displaying order for printing out student information as a result of performing "Search" menu.

Notes

1. There is no deletion menu.
2. We don't consider program's efficiency. The correctness of your program will be mainly considered when grading your program.
3. you may consider using **qsort** function that is defined in `stdlib.h` or **sort** function in STL to implement "Sorting Option".

```
void qsort(void *base,
           size_t num,
           size_t size,
           int (*comp_func)(const void *, const void *))
```