

oop project#1

Student Information Mangement System



Team MEMBER

20181427 김효연

20182338 이승규(PRESENTATION SPEAKER)

20181807 김찬경

20180707 김용덕

20186889 권용한

Contents

1. Project Description
2. Implementation & Description
3. OOP Concepts in our program
4. SW system design(UML diagrams)
5. Execution Results
6. Conclusion

Project Description

Student Management System

Manage name, department, tel, student\_id, birth\_year, total\_credit, total\_avg information

Overview

1. Student.h + Student.cpp - student class definition + function implementation
2. Student\_DB.h + Student\_DB.cpp - DB class definition + function implementation
3. a.cpp - main function
4. file1.txt - txt.file for program maintain

Main Function

1. insert
2. search
3. sort & modify

Additional Functions & Variables

1. using operator
2. modify function
3. total\_credit, total\_avg variables and sort
4. ascending order, descending order

System requirement

made by Visual Studio Compiler in Windows Environment

Implementation & Description

Student.h + Student.cpp

Student class member variables :

1. **string** name, department, tel, student\_id
2. **int** birth\_year, total\_credit, **sort\_type**
3. **double** total\_avg;

Member Function:

1. **getter**

ex) string get\_name() {

return name;

}

1. **setter**

ex) void set\_sort\_type(int sort\_type) {

this->sort\_type = sort\_type;

}

For each variables, we made getter, setter functions

1. **friend ostream& operator<< (ostream& os, Student st);**

ostream& operator<< (ostream& os, Student st) {

os << left << setw(17) << st.name << setw(12) << st.student\_id << setw(20) << st.department << setw(12) << st.birth\_year << setw(13) << st.tel <<setw(16) << st.total\_credit << setw(16) << st.total\_avg;

return os; }

1. Define friend keyword to access private member variables
2. setw(n): Sets the field width to be used on output operations

ex) setw(3) << n // Output: 00n, 00 is blank

1. left : left side sort
2. **bool operator< (Student& st);**

switch (get\_sort\_type()) {

case 1: return this->get\_name() < st.get\_name(); break;

case 2: return this->get\_student\_id() < st.get\_student\_id(); break;

case 3: return this->get\_birth\_year() < st.get\_birth\_year(); break;

case 4: return this->get\_department() < st.get\_department(); break;

case 5: return this->get\_total\_credit() < st.get\_total\_credit(); break;

case 6: return this->get\_total\_avg() < st.get\_total\_avg(); break;

}

Additional Explanation: Implement operation overload for convenient sorting

Ex) if sort\_type==1, we can use sort() function as comparing name function

Student\_DB.h + Student\_DB.cpp

Student\_DB class member variables :

private:

1. **int** index
2. **Student** arr[100]

public:

1. **ifstream** fin
2. **ofstream** fout

Member Functions:

1. **DB()**
2. Constructor
3. if *file1.txt* exists then open, otherwise create the *file1.txt*
4. **init()**
5. if, *file1.txt* exists, read every line in *file1.txt*
6. read line by line and then using substr() to divide information and store the divided information to each member variables

ex) KimHyoYeon 2018123456 Computer science

name = KimHyoYeon / stu\_id = 2018123456 / department = Computer science

1. If there is blank(two times in a row) at the moment of storing, delete the blank

ex) name**000** -> name, 000 is blank

1. At the moment of storing, convert to each data type, and store the data
2. Using stored information, store the student Object at arr[index]
3. **insert\_DB()**
4. Insert each value and store the value at member variables
5. Insert in order of name, student\_id, birth\_year, department, tel, total\_credit, total\_avg
6. Check if there is same value stored when insert student\_id, tel
7. Using checker variable to check repetition, if checker==1, there is same value
8. The length of student\_id should be exactly 10 digits, if not insert until it becomes 10 digits
9. The length of tel should be exactly 12 digits, if not insert until it becomes 12 digits
10. cin.ignore() is used to delete getline’s linefeed
11. Insert total\_credit range of 0 to 120, insert until it gets proper value
12. Insert total\_avg range of 0.0 to 4.5, insert until it gets proper value
13. Using stored information, store the student Object at arr[index]
14. Store arr[index] at file1.txt
15. **search\_DB(int src)**
16. Get input src and the use switch function to move to proper menu
17. if src==1, using name to search, get name and compare the name with Student Object names stored at arr. If there is same name, print student information, otherwise print error message
18. if src==2, using Student ID to search, get ID and compare the ID with Student Object IDs stored at arr. If there is same ID, print student information, otherwise print error message
19. when using Student ID to search, you can modify student information. You could choose by Y or N. if Y is selected, you can modify data by selecting(using switch function) variable(ex. name)
20. ifsrc==3, search using birth\_year
21. if src==4, search using department
22. if src==5, search using total\_credit, get range and print the student information at that range
23. if src==6, search using total\_avg, get range and print the student information at thatrange
24. if src==7, print every student information
25. if src is inserted improperly, print error message
26. **sort\_DB(int sort\_type, int direction)**
27. set all arr’s sort\_type
28. if direction==1, ascending order, if direction==2, descending order
29. execute sort according to sort\_type and direction
30. To prevent errors(if you do "sort" several times or if a file is newly created) that may occur, close the file and reopen it
31. Initialize the file pointer to the beginning.
32. Based on the newly arranged information, overwrite the file
33. **exit\_DB()**

fin, fout close (close the file)

a.cpp (Main)

Variables :

1. **DB** data\_base
2. **int** menu = 0, search = 0, sort\_type = 1(default is name), direction = 1(default is ascending), index;

Sequence :

1. **data\_base.init();**

read information of *file1.txt*

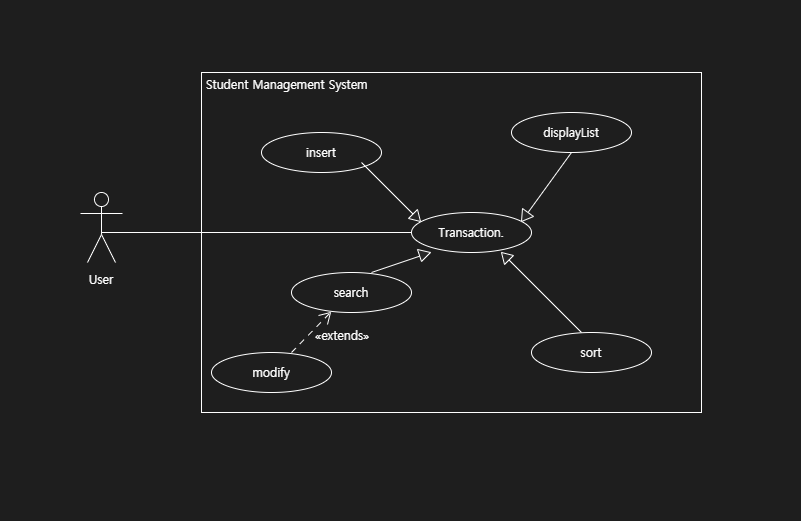
1. **while()**
2. automatically execute sort\_DB(int sort\_type, int direction)
3. display menu and get input
4. if menu==1, execute insert\_DB()
5. if menu==2, get index, if index==0, output a message that says there's no data. / otherwise, display search menu(option) and get input, execute search\_DB(search)
6. if menu==3, get index, if index==0, output a message that says there's no data. / otherwise, display sort menu(option), direction, and get input then change sort\_type, direction variables. The reason why we don’t execute sort is because it runs automatically at 1
7. if menu==4, execute exit\_DB(), End the program(while).

OOP Concepts in our program

1. We create Student class, Student DB class and DB class use Student class for operation
2. By using setter, getter function, we can access to Student class private member variables
3. By using ‘friend’, we directly access to Student class member variables using operator(<<)
4. In position of users, they do not need to know how our program operate when they use our program. They can just input the menu and value to use our program
5. By using operator, we give polymorphism to our sorting operation

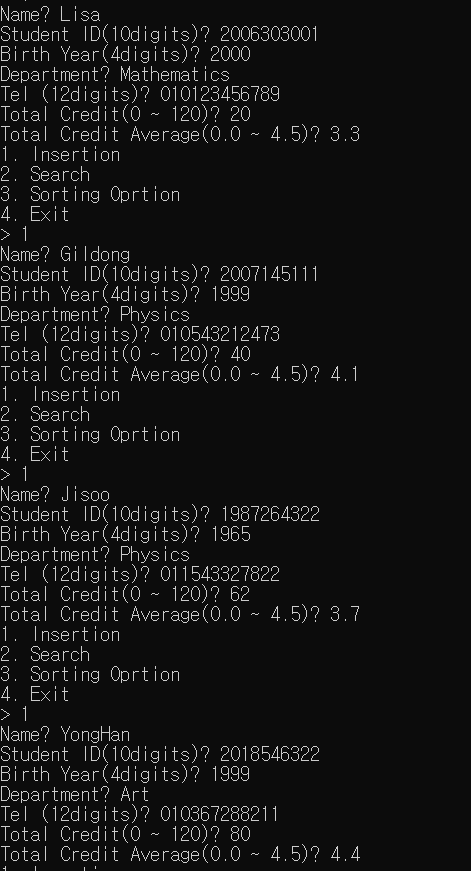
SW System design (UML Diagram)

**class diagram**

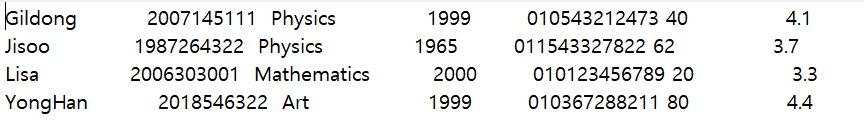
**use-case diagram**

Execution Results

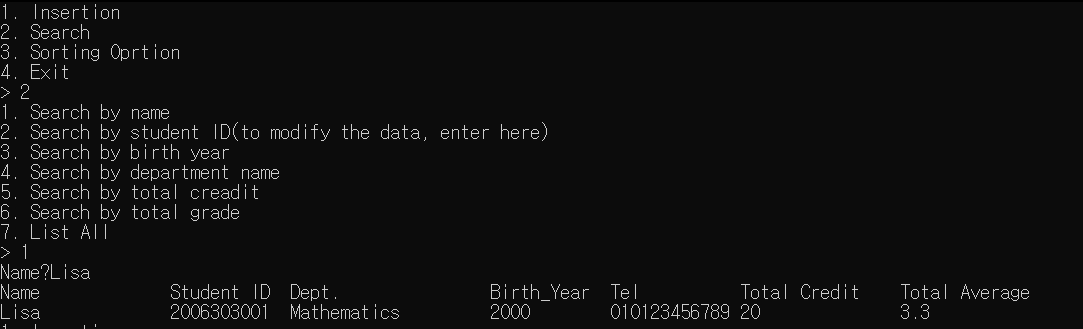
1. **Insertion**



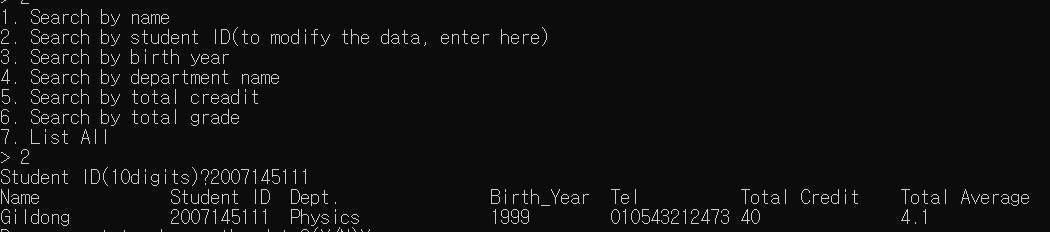
**🔼 executing insertion function**

**🔼 The result of insertion function at the file**

1. **Search**

****

**🔼Execute search by name function and get the proper result**

****

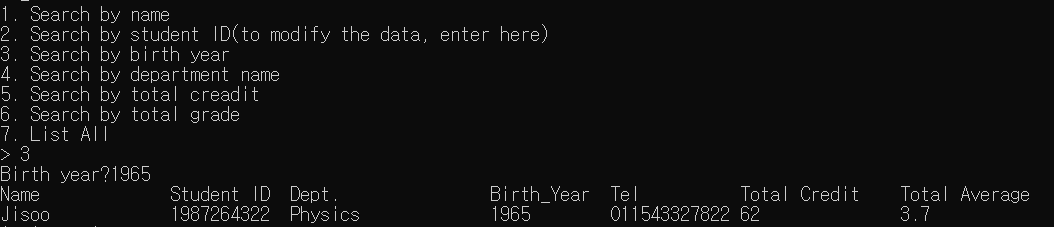
**🔼Execute search by student ID function and get the proper result**

**텍스트이(가) 표시된 사진

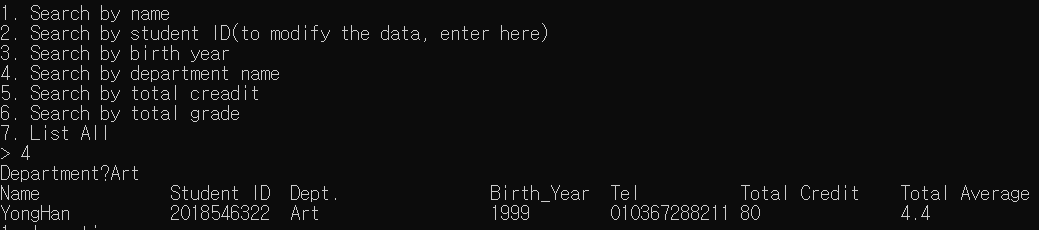
자동 생성된 설명**

**🔼Execute modify function and we can see the name changed**

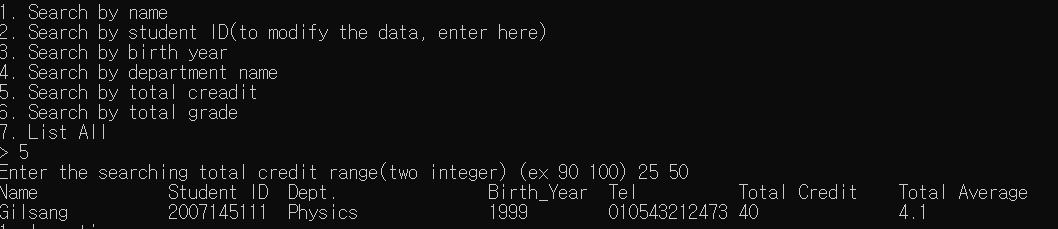
**Gildong to Gilsang**

****

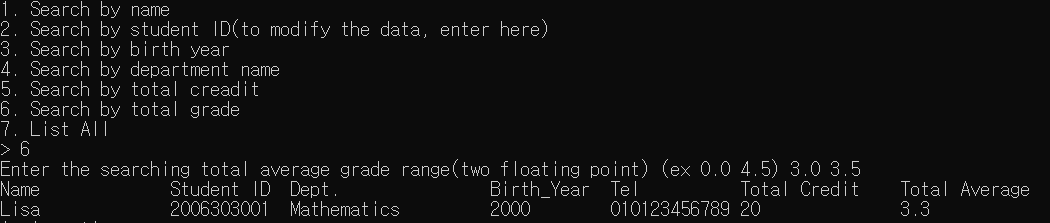
**🔼Execute search by birth year function and get the proper result**

****

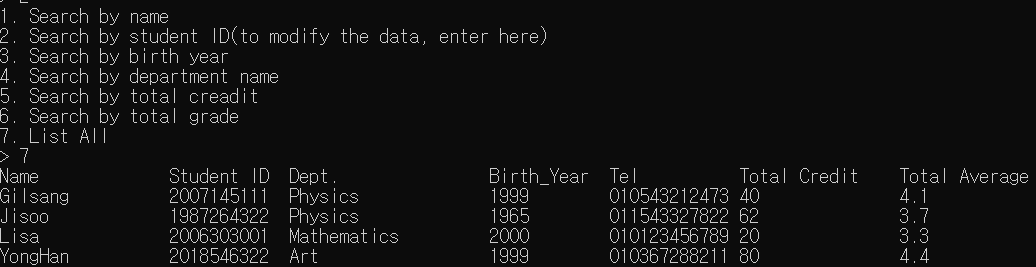
**🔼Execute search by department name function and get the proper result**

****

**🔼Execute search by total credit function and get the proper result**

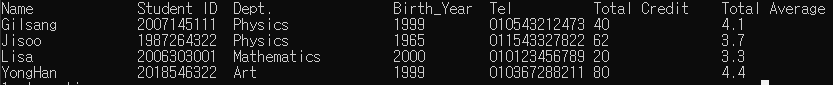
****

**🔼Execute search by total average grade function and get the proper result**

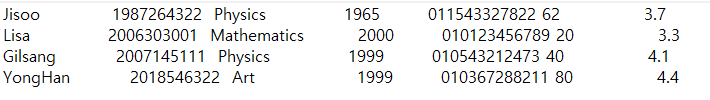
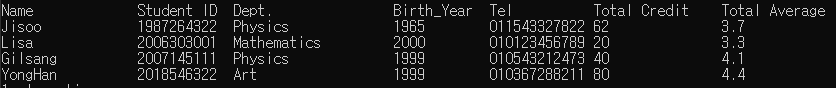
****

**🔼Execute List all function and get the proper result**

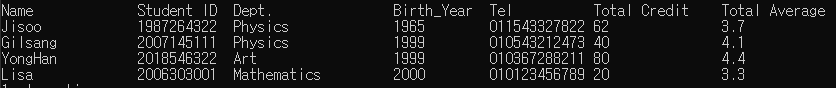
1. **Sorting Option**

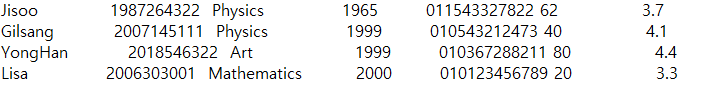
****

**🔼Execute sorting by name, ascending order**

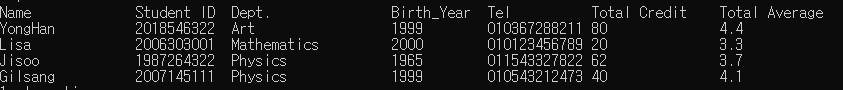
****

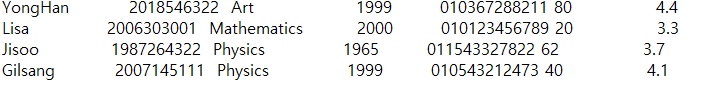
**🔼Execute sorting by student\_id, ascending order**

****

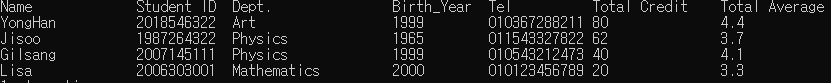
****

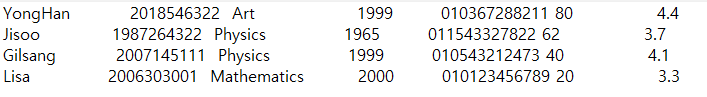
**🔼Execute sorting by birth\_year, ascending order**

****

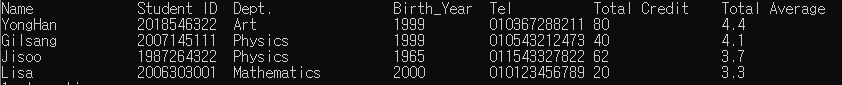
****

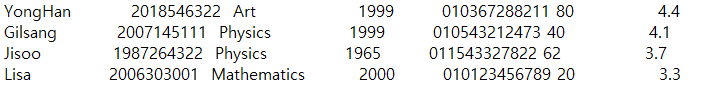
**🔼Execute sorting by department name, ascending order**

****

****

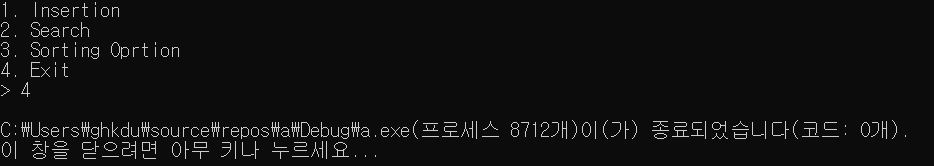
**🔼Execute sorting by total\_credit, descending order**

****

****

**🔼Execute sorting by total\_avg, descending order**

1. **Exit**

****

**🔼Execute Exit function and can see the program ends**

Conclusion

20181427 김효연 :

It was good to learn more about OOP while analyzing the code written by the team members and writing report and PPT

20182338 이승규 :

Before the team project, I was a bit confused of Object Oriented Programming and concepts of class. With this project it was a good chance to handle classes and OOP which was different from C style programming. Also good chance to know how to approach OOP style and how to access and manipulate attributes in class.

20181807 김찬경 :

For this time, I have been using other computer languages when I made projects. So this

project was my first time using C++ to make a project. It was little bit complicated to do the project because I’m not familiar with C++. However, by doing this project, I learned to mediate the opinion between me and other teammate. Also, my ability to make better code advanced.

20180707 김용덕 :

Making C++ project for the first time, I get used to C++ grammar and OOP style. plus, as a team project. it is very worthwhile to me to know that while learn, discuss, check the problem each other, our communication skill rises as well as our C++ skill. I think this experience is useful when I participate in a big project after.

20186889 권용한 :

I have been learned the concepts of OOP, but this time is the first time that I make the OOP project. So at first I think it’s hard to make project but with my teammate we can make program that works well. With this experience, next time I think that I can make better programs.