객체지향 프로그래밍과 자료구조 Lab. 5

5.1 Class Date

```
class Date
{
        friend istream& operator>>(istream&, Date&);
        friend ostream& operator<<(ostream&, const Date&);
public:
        Date();   // default constructor
        Date(int y, int m, int d); // constructor
        void setDate(int newYear, int newMonth, int newDay);
        int getYear() { return year; }
        int getYearDay();
        int getWeekDay();
        int getElapsedDays(); // get elapsed days from AD 1. 1. 1.
        const Date operator=(const Date rightSide);
        bool operator>(Date rightSide);
        bool operator<(Date rightSide);
        bool operator==(Date rightSide);
        bool operator!=(Date rightSide);
        bool isLeapYear(int y); // check whether the given year y is leap year
private:
        bool isLeapYear(); // check whether the year is leap year
        bool isValidDate(int y, int m, int d);
        int year;
        int month;
        int day;
};

bool isLeapYear(int y);
Date genRandDate();
```

5.2 MyString

```
string genRandName();
string genRandDeptName();
```

- genRandName() 함수는 4 ~ 7 문자로 구성되는 학생 이름을 생성하며, 첫 번째 문자는 대문자로 설정
- genRandDeptName() 함수는 3 ~ 4 개의 대문자로 구성되는 학과 코드를 생성

5.3 Class Person

```
class Person
{
        friend ostream& operator<< (ostream&, const Person&);
public:
        Person() { birthDate = Date(0, 0, 0); name = ""; };
        Person(string nm, Date bd) { birthDate = bd; name = nm; };
        void setName(string n) { name = n; }
        void setBirthDate(Date bd) { birthDate = bd; }
        string getName() const { return name; }
        Date getBirthDate() const { return birthDate; }
protected:
        Date birthDate;
        string name;
};
```

5.4 Class Student

```
#include "Person.h"
class StudentArray;
class Student : public Person
{
        friend class StudentArray;
        friend ostream& operator<< (ostream&, const Student&);
public:
        Student();   // default constructor
        Student(int id);
        Student(int id, string n, Date dob, string dept_n, double gpa);
        int getST_id() const { return st_id; }
        string getDept_name() const { return dept_name; };
        double getGPA() const { return gpa; }
        Date getBirthDate() const { return birthDate; }
        void setST_id(int id) { st_id = id; }
        void setDept_name(string dp_n) { dept_name = dp_n; };
        void setGPA(double g) { gpa = g; }
        const Student& operator=(const Student& right);
        bool operator>(const Student& right);
        bool operator==(const Student& right);

private:
        int st_id;
        string dept_name;
        double gpa;
};

Student genRandStudent(int id);
void genST_ids(int num_students, int* st_ids);
```

5.5 Class StudentArray

```
class StudentArray
{
        friend ostream& operator<< (ostream&, const StudentArray&);
public:
        StudentArray(int size); // constructor
        StudentArray(const StudentArray& obj);
        ~StudentArray();
        int size() const { return num_students; }
        Student& operator[] (int index) const;
        void sortByBirthDate();
        void sortByName();
        void sortByST_ID();
        void sortByGPA();
private:
        Student* students;
        int num_students;
        bool isValidIndex(int index) const;
};
```

5.6 main()

(1) genStudent()

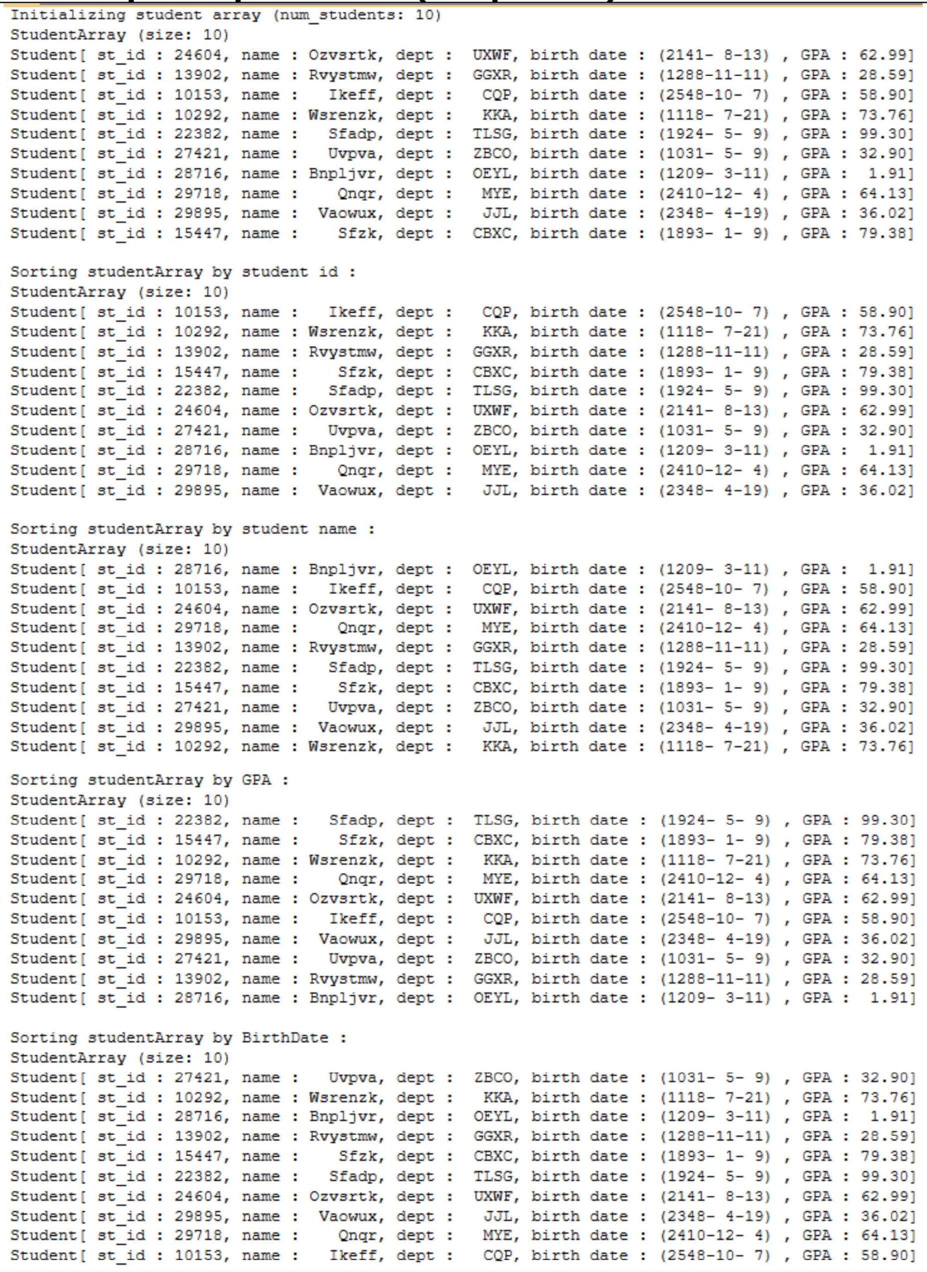
- class Student의 데이터 멤버들을 rand() 함수를 사용하여 생성하고, 이들 데이터 멤버들을 포함하는 class Student 객체를 genRandStudent(int id) 함수를 사용하여 생성

- st_id는 genST_ids(int num_students, int* st_ids) 함수를 사용하여 생성하도록 하고, 10000 ~ 50000 사이의 정수 값을 가지도록 rand() 함수를 사용하여 생성하고, 중복되지 않도록 할 것.
- string 자료형인 name 은 genRandName() 함수를 사용하여 생성하도록 하며, 4 ~ 7 문자로 구성되며, 첫 번째 문자는 대문자로 할 것.
- birthDate는 genRandDate() 함수를 사용하여 생성하도록 하고, 2000. 1. 1. ~ 2999. 12. 31. 내에 있는 날짜를 임의로 선정할 것
- dept_name은 genRandDeptName() 함수를 사용하여 생성하도록 하며, 3 ~ 4 개의 대문자로 구성되는 학과 코드를 생성하여 사용할 것.
- GPA는 0.00 ~ 99.99 사이의 실수를 랜덤으로 생성하여 사용할 것.

```
/* main.cpp (Date, Person, Student, StudentArray) */
#include <iostream>
#include <fstream>
#include "StudentArray.h"
#include <string>
#define NUM_STUDENTS 10

void main()
{
        StudentArray studentArray(NUM_STUDENTS);
        Student st;
        ofstream fout;
        int st_ids[NUM_STUDENTS];

        fout.open("output.txt");
        if (fout.fail())
        {
                cout << "Fail to open an output file (output.txt)\n";
                exit(1);
        }

        genST_ids(NUM_STUDENTS, st_ids);
        fout << "Initializing student array (num_students: " << NUM_STUDENTS << ")" << endl;
        for (int i = 0; i < NUM_STUDENTS; i++)
        {
                st = genRandStudent(st_ids[i]);
                studentArray[i] = st;
        }
        fout << studentArray;

        fout << "\nSorting studentArray by student id : " << endl;
        studentArray.sortByST_ID();
        fout << studentArray;

        fout << "\nSorting studentArray by student name : " << endl;
        studentArray.sortByName();
        fout << studentArray;

        fout << "\nSorting studentArray by GPA : " << endl;
        studentArray.sortByGPA();
        fout << studentArray;

        fout << "\nSorting studentArray by BirthDate : " << endl;
        studentArray.sortByBirthDate();
```

```
            fout << studentArray;

            fout << endl;
            fout.close();
}
```

5.7 Sample Output Result ("output.txt")



```
Initializing student array (num_students: 10)
StudentArray (size: 10)
Student[ st_id : 24604, name : Ozvsrtk, dept :  UXWF, birth date : (2141- 8-13) , GPA : 62.99]
Student[ st_id : 13902, name : Rvystmw, dept :  GGXR, birth date : (1288-11-11) , GPA : 28.59]
Student[ st_id : 10153, name :   Ikeff, dept :   CQP, birth date : (2548-10- 7) , GPA : 58.90]
Student[ st_id : 10292, name : Wsrenzk, dept :   KKA, birth date : (1118- 7-21) , GPA : 73.76]
Student[ st_id : 22382, name :   Sfadp, dept :  TLSG, birth date : (1924- 5- 9) , GPA : 99.30]
Student[ st_id : 27421, name :   Uvpva, dept :  ZBCO, birth date : (1031- 5- 9) , GPA : 32.90]
Student[ st_id : 28716, name : Bnpljvr, dept :  OEYL, birth date : (1209- 3-11) , GPA :  1.91]
Student[ st_id : 29718, name :    Qnqr, dept :   MYE, birth date : (2410-12- 4) , GPA : 64.13]
Student[ st_id : 29895, name :  Vaowux, dept :   JJL, birth date : (2348- 4-19) , GPA : 36.02]
Student[ st_id : 15447, name :    Sfzk, dept :  CBXC, birth date : (1893- 1- 9) , GPA : 79.38]

Sorting studentArray by student id :
StudentArray (size: 10)
Student[ st_id : 10153, name :   Ikeff, dept :   CQP, birth date : (2548-10- 7) , GPA : 58.90]
Student[ st_id : 10292, name : Wsrenzk, dept :   KKA, birth date : (1118- 7-21) , GPA : 73.76]
Student[ st_id : 13902, name : Rvystmw, dept :  GGXR, birth date : (1288-11-11) , GPA : 28.59]
Student[ st_id : 15447, name :    Sfzk, dept :  CBXC, birth date : (1893- 1- 9) , GPA : 79.38]
Student[ st_id : 22382, name :   Sfadp, dept :  TLSG, birth date : (1924- 5- 9) , GPA : 99.30]
Student[ st_id : 24604, name : Ozvsrtk, dept :  UXWF, birth date : (2141- 8-13) , GPA : 62.99]
Student[ st_id : 27421, name :   Uvpva, dept :  ZBCO, birth date : (1031- 5- 9) , GPA : 32.90]
Student[ st_id : 28716, name : Bnpljvr, dept :  OEYL, birth date : (1209- 3-11) , GPA :  1.91]
Student[ st_id : 29718, name :    Qnqr, dept :   MYE, birth date : (2410-12- 4) , GPA : 64.13]
Student[ st_id : 29895, name :  Vaowux, dept :   JJL, birth date : (2348- 4-19) , GPA : 36.02]

Sorting studentArray by student name :
StudentArray (size: 10)
Student[ st_id : 28716, name : Bnpljvr, dept :  OEYL, birth date : (1209- 3-11) , GPA :  1.91]
Student[ st_id : 10153, name :   Ikeff, dept :   CQP, birth date : (2548-10- 7) , GPA : 58.90]
Student[ st_id : 24604, name : Ozvsrtk, dept :  UXWF, birth date : (2141- 8-13) , GPA : 62.99]
Student[ st_id : 29718, name :    Qnqr, dept :   MYE, birth date : (2410-12- 4) , GPA : 64.13]
Student[ st_id : 13902, name : Rvystmw, dept :  GGXR, birth date : (1288-11-11) , GPA : 28.59]
Student[ st_id : 22382, name :   Sfadp, dept :  TLSG, birth date : (1924- 5- 9) , GPA : 99.30]
Student[ st_id : 15447, name :    Sfzk, dept :  CBXC, birth date : (1893- 1- 9) , GPA : 79.38]
Student[ st_id : 27421, name :   Uvpva, dept :  ZBCO, birth date : (1031- 5- 9) , GPA : 32.90]
Student[ st_id : 29895, name :  Vaowux, dept :   JJL, birth date : (2348- 4-19) , GPA : 36.02]
Student[ st_id : 10292, name : Wsrenzk, dept :   KKA, birth date : (1118- 7-21) , GPA : 73.76]

Sorting studentArray by GPA :
StudentArray (size: 10)
Student[ st_id : 22382, name :   Sfadp, dept :  TLSG, birth date : (1924- 5- 9) , GPA : 99.30]
Student[ st_id : 15447, name :    Sfzk, dept :  CBXC, birth date : (1893- 1- 9) , GPA : 79.38]
Student[ st_id : 10292, name : Wsrenzk, dept :   KKA, birth date : (1118- 7-21) , GPA : 73.76]
Student[ st_id : 29718, name :    Qnqr, dept :   MYE, birth date : (2410-12- 4) , GPA : 64.13]
Student[ st_id : 24604, name : Ozvsrtk, dept :  UXWF, birth date : (2141- 8-13) , GPA : 62.99]
Student[ st_id : 10153, name :   Ikeff, dept :   CQP, birth date : (2548-10- 7) , GPA : 58.90]
Student[ st_id : 29895, name :  Vaowux, dept :   JJL, birth date : (2348- 4-19) , GPA : 36.02]
Student[ st_id : 27421, name :   Uvpva, dept :  ZBCO, birth date : (1031- 5- 9) , GPA : 32.90]
Student[ st_id : 13902, name : Rvystmw, dept :  GGXR, birth date : (1288-11-11) , GPA : 28.59]
Student[ st_id : 28716, name : Bnpljvr, dept :  OEYL, birth date : (1209- 3-11) , GPA :  1.91]

Sorting studentArray by BirthDate :
StudentArray (size: 10)
Student[ st_id : 27421, name :   Uvpva, dept :  ZBCO, birth date : (1031- 5- 9) , GPA : 32.90]
Student[ st_id : 10292, name : Wsrenzk, dept :   KKA, birth date : (1118- 7-21) , GPA : 73.76]
Student[ st_id : 28716, name : Bnpljvr, dept :  OEYL, birth date : (1209- 3-11) , GPA :  1.91]
Student[ st_id : 13902, name : Rvystmw, dept :  GGXR, birth date : (1288-11-11) , GPA : 28.59]
Student[ st_id : 15447, name :    Sfzk, dept :  CBXC, birth date : (1893- 1- 9) , GPA : 79.38]
Student[ st_id : 22382, name :   Sfadp, dept :  TLSG, birth date : (1924- 5- 9) , GPA : 99.30]
Student[ st_id : 24604, name : Ozvsrtk, dept :  UXWF, birth date : (2141- 8-13) , GPA : 62.99]
Student[ st_id : 29895, name :  Vaowux, dept :   JJL, birth date : (2348- 4-19) , GPA : 36.02]
Student[ st_id : 29718, name :    Qnqr, dept :   MYE, birth date : (2410-12- 4) , GPA : 64.13]
Student[ st_id : 10153, name :   Ikeff, dept :   CQP, birth date : (2548-10- 7) , GPA : 58.90]
```

<Oral Test>

5.1 객체 지향형 프로그래밍에서 상속 개념을 사용하는 장점은 무엇인가? 예를 들어 설명하라.

5.2 C++ 클래스에서 접근 지정자 “protected”와 “private”의 차이점에 대하여 설명하라.

5.3 상속받은 자식 클래스의 생성자 (constructor method of children class) 에서 상속 받는 부모 클래스의 생성자 (constructor method of parent class)를 호출하는 방법에 대하여 예를 들어 설명하라.

5.4 상속을 사용하는 C++ 클래스에서 상속이 되는 멤버함수와 상속이 되지 않는 멤버함수들을 구분하여 각각 예를 들어 설명하라.