객체지향형 프로그래밍과 자료구조 Lab. 2

2.1 Class Date

1) class Date 정의:

```
/* Date.h */
#ifndef DATE_H
#define DATE_H

#include <iostream>
#include <fstream>
#include <cstdlib>
using namespace std;

#define WEEKDAY_AD01Jan01 MON // the weekday of AD Jan 1.
#define DAYS_PER_WEEK 7
#define Secs_in_Minute 60
#define Secs_in_Hour (Secs_in_Minute * 60)
#define Secs_in_DAY (Secs_in_Hour * 24)
#define LOCAL_GMT_OFFSET_HOUR 9

class Date
{
public:
    Date();   // default constructor
    Date(int y, int m, int d); // constructor
    ~Date(); // destructor
    void inputDate();
    void fprintDate(ostream& fout);
    void setDate(int y, int m, int dy);
    void setRandDateAttributes();
    void setMonth(int m);
    void setYear(int y);
    int getYear() { return year; }
    int getMonth() { return month; } //Returns 1 for January, 2 for February, etc.
    int getDay() { return day; }
    int getYearDay();
    int getYearDay(int m, int d);
    int getWeekDay();
    int getElapsedDaysFromAD010101(); //   get elapsed days from AD 1. 1. 1.
private:
    bool isLeapYear(); // check whether the year is leap year
    bool isLeapYear(int y); // check whether the given year y is leap year
    bool isValidDate(int y, int m, int d);
    int year;
    int month;
    int day;
};
#endif
```

2) 클래스 멤버 함수 구현

- fprintDate(ostream& fout) 함수는 지정된 출력 객체로 class Date 의 속성 (attribute)를 출력
- setRandDateAttribute()는 class Date 의 데이터 멤버들을 class Date 의 속성에 맞는 임의의 초기값들로 설정
- getElapsedDaysFromAD010101()은 서기 1 년 1 월 1 일부터의 누적된 날짜 수를 반환
- isLeapYear()는 해당 년도가 윤년인가를 판단하여 true 또는 false 를 반환

- isValidDate()는 전달된 연월일 값들이 정상적인 class Date 의 속성값들을 가지는가를 확인하고 true 또는 false 를 반환
- class Date 선언은 "Date.h" header 파일에 class Date 의 멤버함수들은 Date.cpp 파일에 구현할 것.

2.2 Class Person

1) class Person 정의

```
/* Person.h */
#ifndef PERSON_H
#define PERSON_H

#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>
#include "Date.h"
using namespace std;

#define MAX_NAME_LEN 15
#define MIN_NAME_LEN 5
class Person
{
public:
    Person() : name(string("nobody")), birthDate(Date(1, 1, 1)) { }
    Person(string n, Date bd) : name(n), birthDate(bd)      {       }
    void setName(string n) { name = n; }
    string getName() { return name; }
    void setBirthDate(Date bd) { birthDate = bd; }
    Date getBirthDate() { return birthDate; }
    void setRandPersonAttributes();
    void fprintPerson(ostream& fout);
private:
    Date birthDate;
    string name;
};
#endif
```

2) 클래스 멤버 함수 구현

- setRandPersonAttributes()는 데이터 멤버들을 class Person 의 속성에 맞는 임의의 값으로 설정. birthDate 는 setRandDateAttributes() 함수를 사용하여 설정.
- fprintPerson(ostream& fout)은 지정된 출력객체로 class Person 의 속성값들을 출력
- class Person 선언은 "Person.h" header 파일에 class Person 의 멤버함수들은 Person.cpp 파일에 구현할 것.

2.3 main() 함수

1) class Date 사용

- <time.h> 라이브러리에서 제공하는 time()과 localtime() 함수를 사용하여 오늘 (today)의 연월일을 파악
- 서기 1 년 1 월 1 일 (AD010101), 올해의 1 월 1 일, 올해의 크리스마스 등의 class Date 객체를 해당 연월일 정보를 사용하여 생성
- 각 날짜 (class Date 의 객체)를 출력하여 정상적으로 class Date 멤버 함수들이 동작하는 가를 확인
- 오늘로부터 올해의 크리스마스까지 남아있는 날짜 수를 계산하여 출력

2) class Person 사용

- class Person 의 객체를 생성할 때 생성자로 초기값을 전달하는 방법과 setOOO() 설정자 함수들을 사용하여 설정하고, getOOO() 함수들을 사용하여 읽고, 출력하는 기능을 시험

- class Person 의 객체 배열을 생성하며, setRandPersonAttribute() 함수를 사용하여 각 데이터 멤버들을 class Person 의 속성에 맞는 초기값으로 설정
- class Person 의 객체 속성들을 fprintPerson() 멤버 함수를 사용하여 출력

```
/* main.cpp */

#include <iostream>
#include <fstream>
#include <time.h> // for time(), localtime(), time_t, struct tm
#include <string>
#include "Date.h"
#include "Person.h"

using namespace std;
#define NUM_PERSON 10

int main()
{
	ofstream fout;
	fout.open("output.txt");
	if (fout.fail())
	{
		cout << "Error in opening output.txt !!" << endl;
		exit;
	}

	/* part 1 - handling class Date */
	Date AD010101(1, 1, 1);
	int year, month, day;
	int daysToChristmas;
	time_t currentTime;
	struct tm *time_and_date;

	time(&currentTime);
	time_and_date = localtime(&currentTime);
	year = time_and_date->tm_year + 1900;
	month = time_and_date->tm_mon + 1;
	day = time_and_date->tm_mday;
	Date newYearDay(year, 1, 1), today(year, month, day);

	fout << "AD Jan. 1, 1 is ";
	AD010101.fprintDate(fout);
	fout << endl;

	fout << "Today is ";
	today.fprintDate(fout);
	fout << endl;

	fout << "New year's day of this year was ";
	newYearDay.fprintDate(fout);
	fout << endl;

	Date christmas(year, 12, 25);
	fout << "Christmas of this year is ";
	christmas.fprintDate(fout);
	fout << endl;

	if (today.getMonth() == christmas.getMonth() &&
		today.getDay() == christmas.getDay())
```

```
        {
            fout << "Today is Christmas!   Happy Christmas to you all !!◘";
        }
        else {
            fout << " Sorry, today is not Christmas!◘";
            daysToChristmas = christmas.getElapsedDaysFromAD010101() -
          today.getElapsedDaysFromAD010101();
            fout << " You must wait " << daysToChristmas << " day(s) to Christmas !" << endl;
        }
        fout << endl;

        /* part 2 - handling class Person */
        Person p1(string("Hong, Gil-Dong"), Date(2000, 1, 1)), p2;
        fout << "Person p1 : " << endl;
        p1.fprintPerson(fout);
        fout << endl;

        fout << "Person p2 : " << endl;
        p2.setName(string("Lee, Soo-Jeong"));
        p2.setBirthDate(Date(2018, 9, 1));
        p2.fprintPerson(fout);
        fout << endl;

        fout << endl << "Generating array of persons ... " << endl;
        Person *persons;
        persons = (Person *) new Person[NUM_PERSON];
        for (int i = 0; i < NUM_PERSON; i++)
        {
            persons[i].setRandPersonAttributes();
            persons[i].fprintPerson(fout);
            fout << endl;
        }
        fout << endl;
        delete[] persons;
        fout.close()
        return 0;
}
```

2.4 Execution Result

```
AD Jan. 1, 1 is    January  1,    1 (    Monday)
Today is  September  4, 2021 (  Saturday)
New year's day of this year was    January  1, 2021 (    Friday)
Christmas of this year is   December 25, 2021 (  Saturday)
 Sorry, today is not Christmas!
 You must wait 112 day(s) to Christmas !

Person p1 :
 Person [name: Hong, Gil-Dong  , birth date:    January  1, 2000 (  Saturday)]
Person p2 :
 Person [name: Lee, Soo-Jeong  , birth date:  September  1, 2018 (  Saturday)]

Generating array of persons ...
 Person [name: Humea           , birth date:   December 11, 1041 (  Saturday)]
 Person [name: Fdxfir          , birth date:       June  6, 1464 (    Monday)]
 Person [name: Xggbwkf         , birth date:      April 25, 2436 (    Friday)]
 Person [name: Xwfnfozvsrtkj   , birth date:       July 16, 2447 (   Tuesday)]
 Person [name: Ggxrpnrvystm    , birth date:       June 29, 2711 (  Thursday)]
 Person [name: Yycqp           , birth date:    January  6, 2842 (    Monday)]
 Person [name: Effmz           , birth date:    October  5, 2890 (  Thursday)]
 Person [name: Kasvwsrenzk     , birth date:    January 22, 1623 (    Sunday)]
 Person [name: Xtlsgy          , birth date:       July 21, 1118 (    Sunday)]
 Person [name: Dpooefxzbco     , birth date:      March 23, 2977 (    Sunday)]
```

<2020-2 객체지향형 프로그래밍과 자료구조 실습 2 Oral Test>

학번		성명		점수	

(1) C++ 객체 지향형 프로그래밍에서 캡슐화 (encapsulation)이란 무엇이며, 어떤 장점이 있는가를 예를 들어 설명하라. (핵심포인트: 캡슐에 포함되는 항목, 캡슐화에 따른 데이터 추상화, 정보 은닉, 정보 보호)

(2) C++ 객체 지향형 프로그래밍에서 외부 접속 (external interface)과 내부 구현 (internal implementation)을 분리한다는 것이 무엇이며 어떤 장점이 있는가를 예를 들어 설명하라. (핵심포인트: 왜 접속과 구현을 분리하는 것이 좋은가?)

(3) C++ 객체 지향형 프로그래밍에서 정보 보호 (information protection)은 어떻게 하는지 예를 들어 설명하라. (핵심 포인트: 정보보호를 어떻게 하는가?)

(4) C++ 객체 지향형 프로그래밍에서 정보 은닉 (information hiding)은 어떻게 하며, 왜 필요한가에 대하여 예를 들어 설명하라. (핵심 포인트: 정보은닉을 어떻게 하며, 왜 해야 하는가?)