

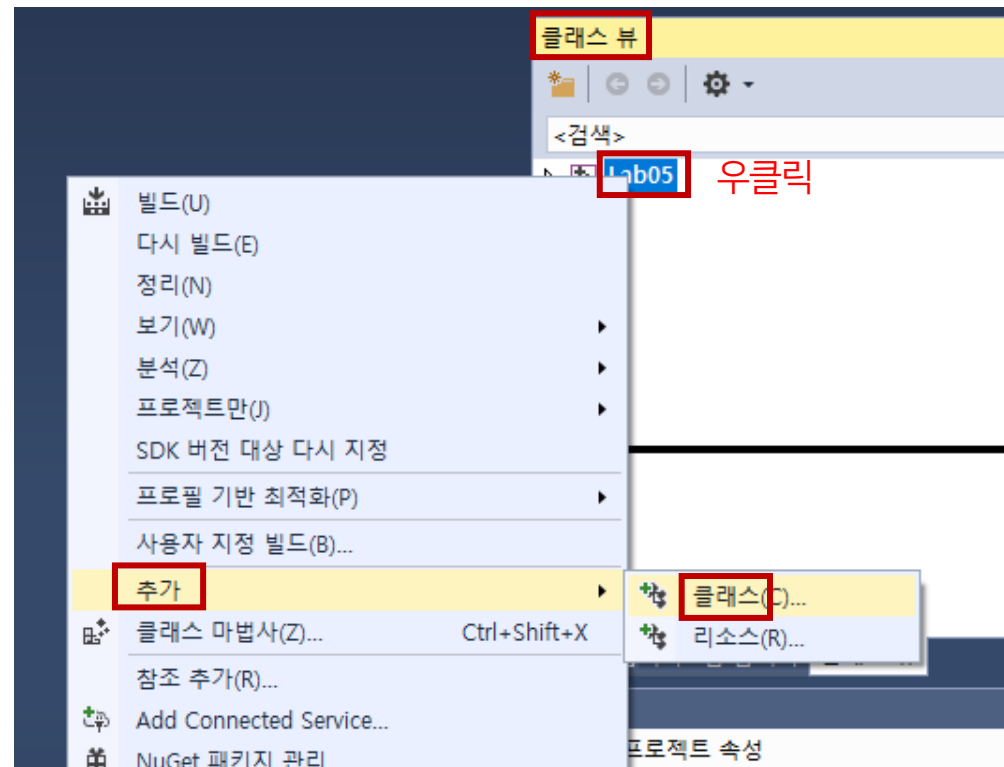
윈도우프로그래밍 Lab 05

- 클래스 Set 완성하기 -

김한슬
uo3359@sookmyung.ac.kr

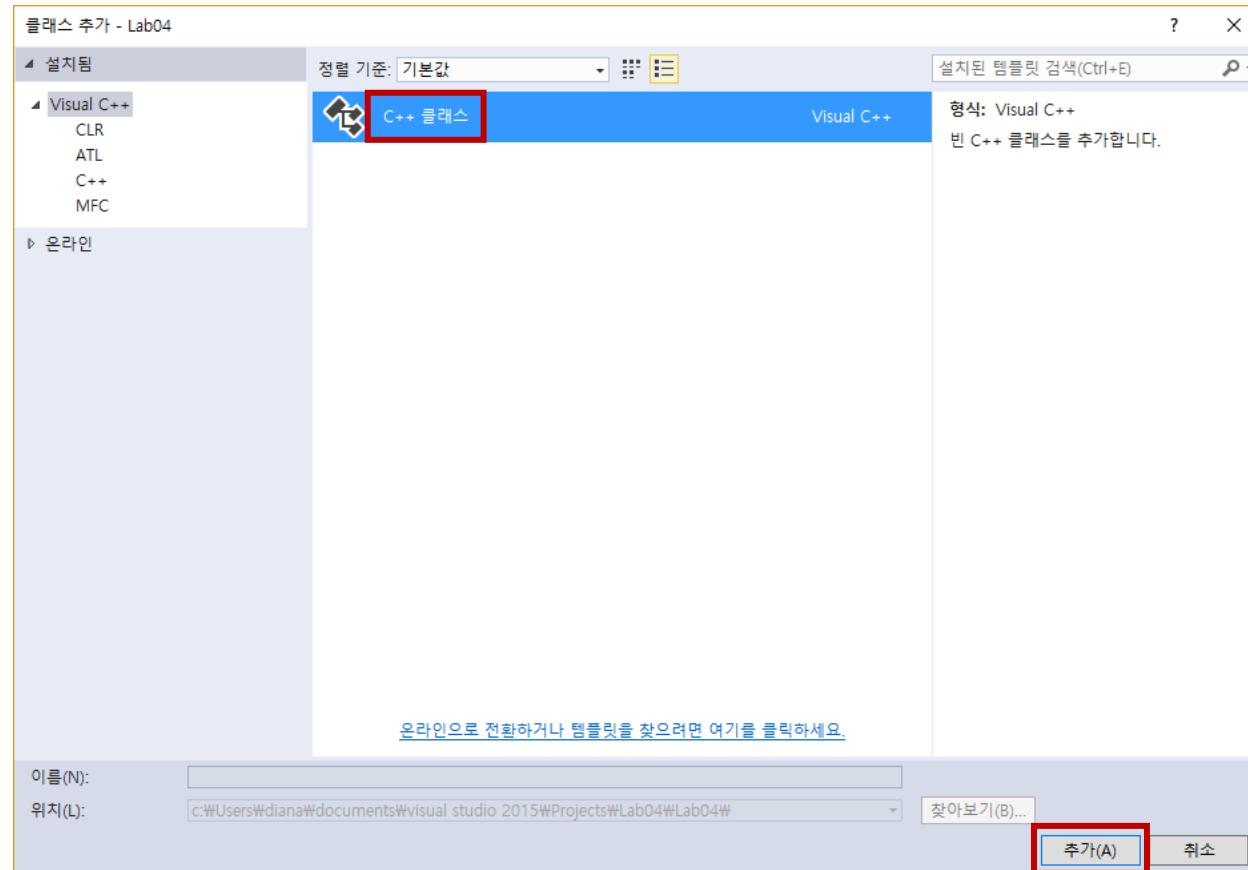
visual studio 2015에서 클래스 추가

- [클래스 뷰] 에서 “프로젝트 이름” 우클릭 -> 추가 -> 클래스



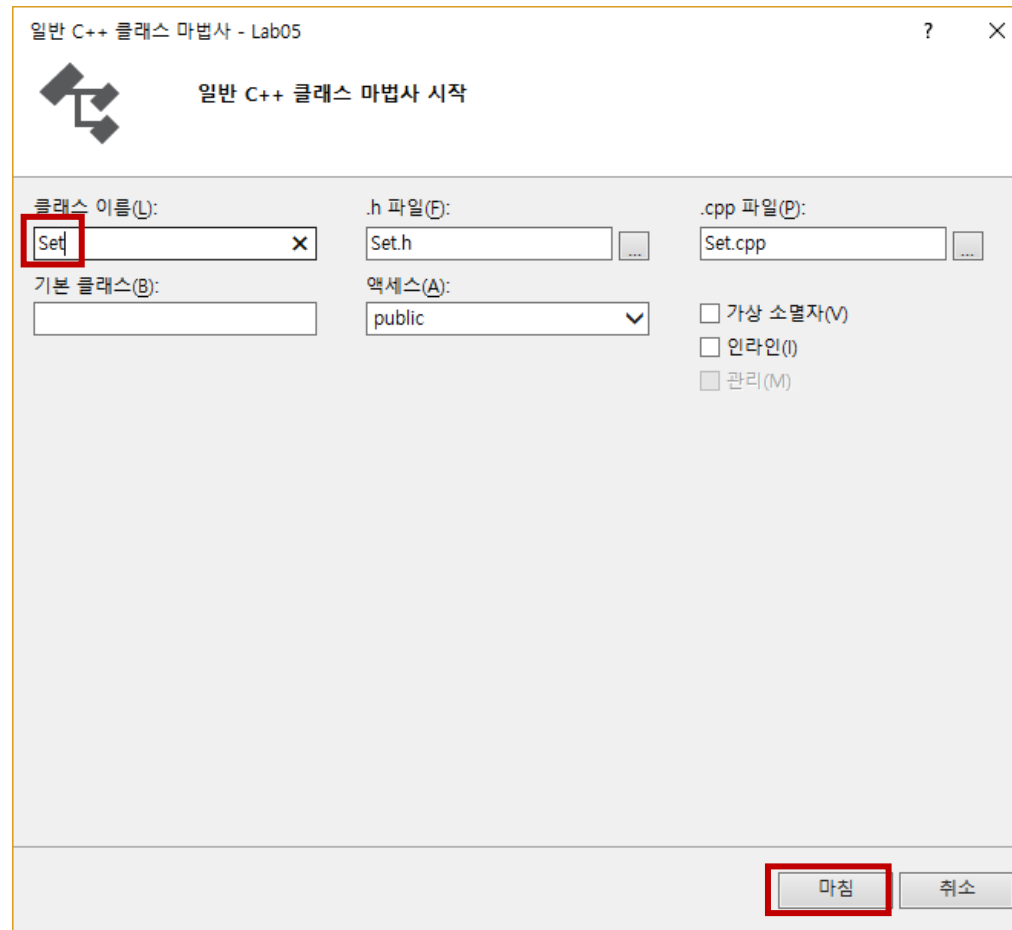
visual studio 2015에서 클래스 추가

- [클래스 추가] 대화상자 -> [C++ 클래스] 선택 -> 추가



visual studio 2015에서 클래스 추가

- [일반 C++ 클래스 마법사] -> 클래스 이름 "Set" -> 마침



과제

1. Set 클래스 선언 (Set.h)

- Set.h 코드 복사 하여 사용
- Set.h 는 변경하지 말 것

과제

2. 멤버 함수 정의 완성하기 (Set.cpp)

- Set.cpp 코드 복사하여 사용
- 기존에 구현된 내용은 변경하지 말 것
- **//TO BE COMPLETED** 부분에 코드를 추가하여 기능을 완성함
- 연산자 오버로딩 참고 (교재 P.42 - 44)

과제

3. main() 함수를 포함할 C++ 소스 파일 (Main.cpp)

- Main.cpp 코드 복사하여 사용
- 기존에 구현된 내용은 변경하지 말 것
- **//TO BE COMPLETED** 부분에 코드를 추가하여 실행 결과 예제와 같이 출력되도록 완성함
- 오버로딩된 연산자는 함수처럼 호출될 수도 있지만, 원래 연산자처럼 식의 표현에 사용하는 것이 일반적인 용법이다. 예를 들면, + 연산자가 오버로딩된 경우, operator+(a, b)처럼 호출될 수도 있지만, a + b와 같이 사용하는 것이 원래의 취지를 살린 용법이다. (교재 p.42 참고)

과제

```
#pragma once

#define MAXSIZE 100 // 집합의 최대 크기

class Set {
    int size; // 집합의 크기
    int elems[MAXSIZE]; // 집합 원소
public:
    Set(void); // default constructor creates an empty set
    Set(Set& s); // copy constructor ((*this) == s)
    Set(int a[], int n); // creates a set from an array
    int getsize(); // return set size
    int get(int i); // return the element indexed by i;
    Set operator+(Set s); // set union
    Set operator-(Set s); // set difference
    Set operator=(Set s); // set assignment; return the assigned set
    int operator>(Set s); // proper subset relation test
    int operator>=(Set s); // subset relation test
    int operator==(Set s); // set equality test
    int operator>>(int x); // element membership test
};
```


과제

```
#include <iostream>
using namespace std;
#include "Set.h"

void setprint(Set s)
{
    cout << "{";
    for (int i = 1; i <= s.getsize(); i++)
        cout << s.get(i) << ((i < s.getsize()) ? ", " : "");
    cout << "}";
}

void main()
{
    int a[] = {3, 9, 5};
    int b[] = {2, 4, 5, 8, 10};
    Set s1;
    Set s2(a, 3);
    Set s3 = s2; // copy constructor is used here
    Set s4(b, 5);

    cout << "SET UNION TEST" << endl;
    s1 = s3 + s4;
    cout << "s3 = "; setprint(s3); cout << endl;
    cout << "s4 = "; setprint(s4); cout << endl;
    cout << "s3 + s4 = "; setprint(s1); cout << endl << endl;

    // s1 = s4 - s3
    // TO BE COMPLETED

    // test for i = 1, 2, ..., 10 if i is in s4
    // TO BE COMPLETED

    // test if s4 < s1
    // TO BE COMPLETED

    // test if s2 == s3
    // TO BE COMPLETED
}
```

실행 결과

```
C:\Windows\system32\cmd.exe

SET UNION TEST
s3 = {3, 9, 5}
s4 = {2, 4, 5, 8, 10}
s3 + s4 = {3, 9, 5, 2, 4, 8, 10}

SET DIFFERENCE TEST
s4 = {2, 4, 5, 8, 10}
s3 = {3, 9, 5}
s4 - s3 = {2, 4, 8, 10}

SET MEMBERSHIP TEST
s4 = {2, 4, 5, 8, 10}
1 is NOT in s4
2 is in s4
3 is NOT in s4
4 is in s4
5 is in s4
6 is NOT in s4
7 is NOT in s4
8 is in s4
9 is NOT in s4
10 is in s4

PROPER SUBSET RELATION TEST
s1 = {2, 4, 8, 10}
s4 = {2, 4, 5, 8, 10}
s1 is a proper subset of s4

SET EQUALITY TEST
s2 = {3, 9, 5}
s3 = {3, 9, 5}
s2 is equal to s3
계속하려면 아무 키나 누르십시오 . . .
```

과제 제출: 보고서 + 솔루션 폴더

- 내용
 - 각 과제의 프로그램 소스 및 설명
 - 각 과제의 실행 화면 캡처
- 제출 형식
 - 스노우보드에서 Lab05_이름_학번.doc 다운로드
 - 스노우보드에 제출: Lab05_이름_학번.zip (아래 두 파일 반드시 포함하도록)
 1. 소스 코드 : Lab05_이름_학번 (압축하지 않은, 모든 디버그 폴더가 삭제 된 솔루션 폴더)
 2. 보고서: Lab05_이름_학번.doc (설명 없을 경우 감점)
- 제출 기한
 - 4월 13일(화) PM 11:59 까지

검사 항목

1. 폴더 / 파일 삭제

- 솔루션 폴더에서 Debug 폴더 삭제
 - visual studio 2010의 경우 ipch 폴더와 .sdf 파일도 삭제해야 함
- 프로젝트 폴더에서 Debug 폴더 삭제

2. 소스 파일

- Set.h 파일을 변경하지 말아야 함
- Set.cpp, main() 함수의 기존에 구현된 내용을 변경하지 말아야 함

3. 컴파일(빌드) 가능해야 함

4. 실행 결과가 잘 출력되어야 함 (각 항목별 부분 채점)