



C++ STL Programming

Project 과제

□ 과제 배점

50점 만점

[참고] Quiz : 20점, Final Exam : 30점, Project : 50점

□ 과제 설명

총 5개의 프로그래밍 과제 입니다. 일부 과제만 해결시 부분 점수가 적용됩니다.

□ 제출 방법

“영문이름_사번” 으로 폴더를 만들고, 폴더안에

과제설명.txt 을 만들고

- (1) 몇 번 과제를 해결했는지 적어 주세요..
- (2) 어떤 컴파일러를 사용했는지를 적어 주세요

해결한 과제만, “**project1.cpp, project2.cpp, project3.cpp** ...” 으로 소스를 만들어 주세요. 실행파일은 필요 없습니다. 소스만 포함해 주시면 됩니다. 소스 파일의 이름 규칙을 꼭 지켜 주세요(대소문자 포함)

폴더를 포함해서 통째로 압축해서 LGE MOOC “Final Project”에서 “select file” 버튼을 눌러서 압축파일을 올려 주세요

□ 제출 기한

과정 종료일 까지(Syllabus 참고)

과제 1. STL의 `copy_if` 알고리즘을 사용하면 주어진 구간에서 조건을 만족하는 요소를 다른 컨테이너로 복사 할 수 있습니다. `copy_if` 와 동일하게 동작하는 `xcopy_if` 를 만들어 보세요.

```
#include <iostream>
using namespace std;

int main()
{
    int x[10] = { 1,2,3,4,5,6,7,8,9,10 };
    int y[10] = { 0 };

    xcopy_if(x, x + 10, y, [](int a) { return a % 2 == 0; });

    for (auto n : y)
        cout << n << endl;
}
// 출력 결과 : 2, 4, 6, 8, 10, 0,0,0,0,0
```

template 파라미터를 받을 때, T1, T2 등을 사용하지 말고, InputIter 처럼 반복자 카테고리를 구별할 수 있도록 표기해 주세요.

과제 2. 아래와 같이 출력 결과가 나올수 있도록 ? 를 채워 보세요

```
#include <iostream>
#include <algorithm>
using namespace std;

int main()
{
    int x[10] = { 1,2,3,4,5,6,7,8,9,10 };
    int y[10] = { 0 };

    copy_if( ? );

    for (auto n : y)
        cout << n << endl;
}
// 출력 결과 : 0, 0, 0, 0, 0, 10, 8, 6, 4, 2
```

과제 3. 아래 main 함수의 주석의 내용을 구현해 보세요.

```
struct People
{
    string name;
    int age;

    People(string n, int a) : name(n), age(a) {}
    void Dump() const { cout << name << "(" << age << ")" << endl; }
};

int main()
{
    vector<shared_ptr<People>> v;

    // ("kim", 10), ("lee", 15), ("park", 5), ("jung", 20), ("choi", 12) 를 v에 넣어
    // 주세요.

    // 이름순으로 정렬해서 출력해 보세요

    // 나이순으로 정렬해서 출력해 보세요.
}
```

비교 정책을 전달하는 방법은 어떠한 방법을 사용해도 상관없습니다.

shared_ptr 객체를 만들 때 는 make_shared를 사용해 주세요.

과제 4. 전처리기 만들기

“preprocessor” 라는 실행파일을 만드세요.

input.txt 파일이 아래와 같을 때

```
define MAX 10
define MIN 5

n1 = MAX
n2 = MIN
```

preprocessor input.txt result.txt

처럼 실행 하면 다음과 같은 result.txt 가 생성되어야 합니다.

```
n1 = 10
n2 = 5
```

unordered_map 을 사용해 주세요.

과제 제출시 실행파일이 아닌 preprocessor 의 소스를 제출하시면 됩니다.

과제 5. 통보 센터 만들기

6-4강에서는 NotificationCenter를 만들었습니다. 동영상 강좌에서 만든 코드는 통보센터에 등록된 순서대로 함수가 호출 됩니다.

NotificationCenter 코드를 수정해서 함수 등록시 우선순위를 전달할 수 있고, 멀티 스레드가 지원되도록 수정해 보세요.

```
void foo(void* p, string name)
{
    cout << name << " : " << this_thread::get_id() << endl;
    this_thread::sleep_for(3s);
    cout << name << " end" << endl;
}

int main()
{
    NotificationCenter nc;

    // Register 의 3번째 인자는 우선순위, 4번째 인자는 스레드사용여부 입니다.
    // 우선순위 1일 제일 낮고, 3일 제일 높다가 가정합니다.
    // NEW_THREAD 지정시 해당 함수는 새로운 스레드로 동작해야 합니다.
    nc.Register("TEST_EVENT", bind(&foo, _1, "one"s), 1);
    nc.Register("TEST_EVENT", bind(&foo, _1, "two"s), 2, NEW_THREAD);
    nc.Register("TEST_EVENT", bind(&foo, _1, "three"s), 3);

    nc.Notify("TEST_EVENT", (void*)0);    // 등록된 순서가 아닌 우선순위가
                                           // 높은 순서로 실행되게 해주세요
                                           // NEW_THREAD 옵션을 가진 경우. 새로운
                                           // 스레드로 해당 함수를 동작되게 해주세요.

    getchar();
}
```

스레드 생성은 C++ 표준 라이브러리를 사용해 주세요. (thread 또는 async)