

---

## STL - 1주 1일

*Wulong*

1. 과목 소개
2. 컴파일 환경
3. 시작 퀴즈

## 1. 과목소개

수강생 여러분. 안녕하세요.

이렇게 글로 인사하게 되니 새로운 맛이 있습니다.

이 과목은 3학년 대상이어서 많은 학생들이 C++ 언어를 수강했을 것이고 내가 어떤 식으로 강의를 하는지 알고 있을 거라 생각합니다.

지금까지 강의록 없이 강의하는 것을 원칙으로 수업을 해 오다 보니 준비된 강의자료는 당연히 없습니다. 빨리 직접 강의를하기를 바랍니다만 일단 2주는 이렇게라도 자료를 준비하며 지낼 수 밖에 없습니다.

C++ 언어의 발전 속도는 매우 빠릅니다. 그래서 강의하는 사람도 공부하지 않으면 지난 이야기나 늘어놓게 됩니다. 변화의 속도가 워낙 빠르다 보니 적절한 강의교재를 찾기도 쉽지 않으며 이전의 이름난 교재는 이제는 표준에서 사라진 내용을 담고 있거나 혁신적인 언어의 변화를 전혀 반영하지 못하고 있습니다.

작년에 이어 이번 학기에도 **교재로 제시**하는 “C++14 STL 철저입문”은 C++17을 넘어 올해 C++20 표준이 확정되는 시점에 비추어 너무 이전 내용이 아닌가 생각 될 것입니다. 이 책의 의의는 현대 C++ STL을 체계적으로 잘 설명하고 있다는 점에 있습니다. 잘 학습한다면 추후 변경된 내용을 스스로 보완해 나가는데 큰 어려움이 없을 정도로 핵심 내용을 서술하고 있는 책입니다.

물론 여러분 모두 잘 아시겠지만 내 강의는 책의 내용을 따라 가지 않습니다. 책에 있는 지식을 다시 설명한다는 것은 교수의 역할이 아니라고 나는 생각합니다. 지금까지 그렇게 강의했듯이 강의계획서에 명시한 줄거리를 따라

- 왜 이러한 내용이 의미있는 것인지
- 어떻게 만들어지게 되었는지
- 어디에 응용하는 것인지
- 앞으로 어떻게 발전해 나갈 것인지

와 같은 내용을 강의하겠습니다.

책이 있는 학생은 책을 먼저 보고 책이 없는 학생은 구글, 유튜브, 블로그 등에서 관련 내용을 찾아 공부하면 되겠습니다.

STL로 바로 들어가기 전에 여러분은 C++, 자료구조, 알고리즘을 알고 있어야 합니다.

1학년/2학년에 배웠던 C++ 언어는 현실에 존재하는(혹은 가상의) 객체를 모델링하기 위한 객체지향언어(Object Oriented Programming Language)로서의 기능이었습니다. 이때 공부했던 것을 간단하게 정리하면 다음과 같은 내용들이었습니다.

- 키워드 `class`를 사용한 객체의 모델링
- 객체의 속성과 행동을 하나의 자료형으로 표현
- 객체를 기존 자료형과 동일한 방식으로 사용할 수 있게 하는 연산자 오버로딩
- 객체와 다른 객체와의 관계
- 한 객체와 다른 객체를 특별한 방식으로 엮는 상속
- 키워드 `virtual`을 사용한 다형성 구현

이번 학기에 공부하게 될 STL(표준 템플릿 라이브러리)은 객체지향이라는 개념의 완전 반대편에 있는 **Generic Programming**(일반적인 프로그래밍)이라는 개념을 구현한 C++ 언어의 표준 라이브러리입니다. 템플릿으로 작성된 코드이며 자료구조와 알고리즘을 반복자라는 디자인 패턴으로 엮은 C++ 언어의 핵심입니다. 내용이 방대하기 때문에 한 학기에 이것을 다 설명하는 것은 매우 어렵습니다.

C++ 언어만 공부하는 나는 STL의 핵심을 학생들에게 흥미롭게 설명할 자신이 있습니다. 그렇지만 이렇게 정리된 자료를 제시하는 것은 큰 벽과 같으며 내가 지향하는 강의가 아니라는 점을 밝힙니다. 강의는 학생들과 강의실에서 서로 이야기하며 함께 하는 것이 맞습니다.

자. 이제 잔소리는 이정도하고 본격적인 강의를 시작하기 전에 C++ 복습을 하면서 기억을 되살리도록 하겠습니다. 또 강의를 위한 코드를 먼저 학습하도록 하겠습니다. 강의 내용을 정리해 볼 수 있도록 간단한 과제를 내고 보고서를 받도록 하겠습니다.

복습할 것과 새로 배울 필요한 내용은 다음과 같습니다.

- 클래스 복습 (자료를 따로 올리겠습니다)
- 템플릿 복습
- 파일 입출력
- 알고리즘의 시간을 재는 코드
- 강의 저장을 위한 `save` 함수 만들기

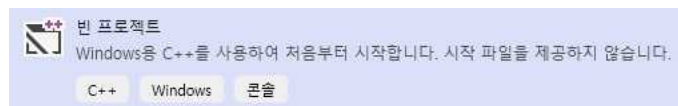
모르는 내용은 책이나 인터넷 검색을 통해 꼭 복습하여야 합니다.

## 2. 컴파일 환경

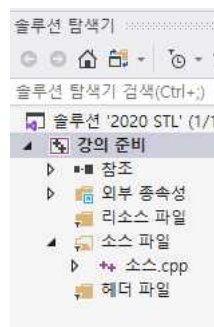
비주얼 스튜디오를 사용합니다. 그림은 무료 다운로드할 수 있는 VS 2019 버전입니다.



이번 학기에 사용할 프로젝트 형식은 다음 그림과 같습니다.



솔루션 탐색기에서 "소스.cpp"를 추가합니다.

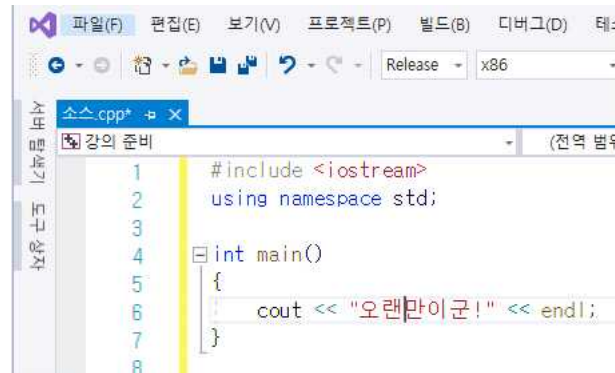


이렇게 코딩해 봅니다.

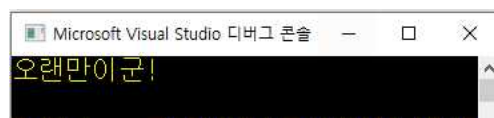
```
#include <iostream>
using namespace std;

int main()
{
    cout << "오랜만이군요!" << endl;
}
```

설마 이 코드를 어디에 작성해야 하는 지 모르는 학생은 없겠죠! 그래도 혹시 모르니 그림과 같은 지 확인해 봅시다. **Release**도 확인했나요?



이제 실행시켜 결과를 봅니다.



항상 실행 결과는 콘솔 창으로 확인하겠습니다.

간단하게 프로그래밍 환경을 정리해 봤습니다.

이제 프로그램을 시작해 보겠습니다.

### 3. 시작 퀴즈

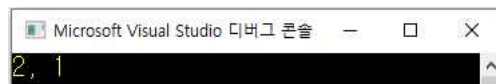
```
#include <iostream>
using namespace std;

int main()
{
    int a { 1 };
    int b { 2 };

    change( a, b );

    cout << a << ", " << b << endl;
}
```

[문제] 실행 결과가 다음과 같이 나오도록 함수 change를 선언하고 정의하는 문제입니다. main 내부의 코드는 한 글자도 변경하면 안됩니다. C언어는 이렇게 할 수 없지만 C++언어는 이렇게 할 수 있습니다.



내 수업은 언제나 컴퓨터를 사용하며 필요한 내용을 검색하며 진행합니다. 시험을 볼 때도 같습니다. 얼른 위 코드를 타이핑해 보세요.

```
int b { 2 };

change( a, b );

cout << a << ", " << b << endl;
```

change에 빨강 밑줄이 있네요. 고쳐주세요.

안타깝습니다. 적절한 시간을 쓰며 생각해 본 후 해결책을 제시하는 과정을 글로 표현하려니 쉽지가 않습니다. 그냥 답을 제시하는 것은 배움에 큰 도움이 되지 않습니다. 동적인 강의를 이렇게 정적을 표현하려니 답답합니다. 이것은 이 강좌를 녹화한다고 해서 해결될 것도 아닙니다. 뭐 그냥 해 보는 소리입니다.

강의를 하게 되면 그때 내 식대로 하기로 하고 지금은 일단 답을 보이겠습니다.

```

#include <iostream>
using namespace std;

void change( int&, int& );           // 함수의 선언

int main()
{
    int a { 1 };
    int b { 2 };

    change( a, b );

    cout << a << ", " << b << endl;
}

void change( int& a, int& b )       // 함수의 정의
{
    int t { a };

    a = b;

    b = t;
}

```

함수의 선언과 정의를 어디에 어떻게 하는 지 보세요. 코딩 스타일도 살펴보세요. 깔끔해 보이나요? 읽고 이해하기 쉽게 코딩하도록 신경을 써 보세요.

이 코드에서 핵심내용은 레퍼런스(reference) 입니다. 관련 내용을 복습하세요. 여기에서 중요 질문을 해 봅니다. 어떻게 대답하겠습니까? 1학년 학생이 물어 본다고 생각하고 답 해보세요.

**[질문] 레퍼런스를 왜 사용합니까? 어디에 사용합니까?**

내가 생각하는 답을 여기에 적지는 않겠습니다. 각자 공부한 내용을 적어보세요. 나중에 수업 시간에 질문해 주세요. 내가 어떻게 생각하고 설명하는지를 그때 봐 주세요.

위의 코드를 이해 했다면 1일차 강의를 다음 문제를 제시하며 마치도록 하겠습니다. 충분한 공부가 될 겁니다. 이 문제의 답은 1주 강의의 과제 리포트에 제출해 주세요.

다음은 과제 코드입니다.

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

int main()
{
    X a { 1 };
    X b { 2 };

    change( a, b );

    cout << a << ", " << b << endl;
}
```

앞의 코드와 같이 두 객체의 값을 서로 바꾸는 프로그램입니다. 어디가 다른 지 금방 찾을 수 있을 겁니다. 이 코드가 실행되면 이전과 같이 화면에 2, 1이 출력되어야 합니다. 또한 main의 내용을 수정하면 안 되는 것도 같습니다.

#### [1주 1일 - 과제 ]

[문제 1] 클래스 X를 정의하라.

[문제 2] change 함수를 선언하고 정의하라.

1주 1일과 1주 2일 과제를

문서 파일 하나로 만들어 (형식은 윈도우 텍스트 문서 형식으로)

2주 1일 강의 시작 전까지 e-class로 제출해 주세요.