

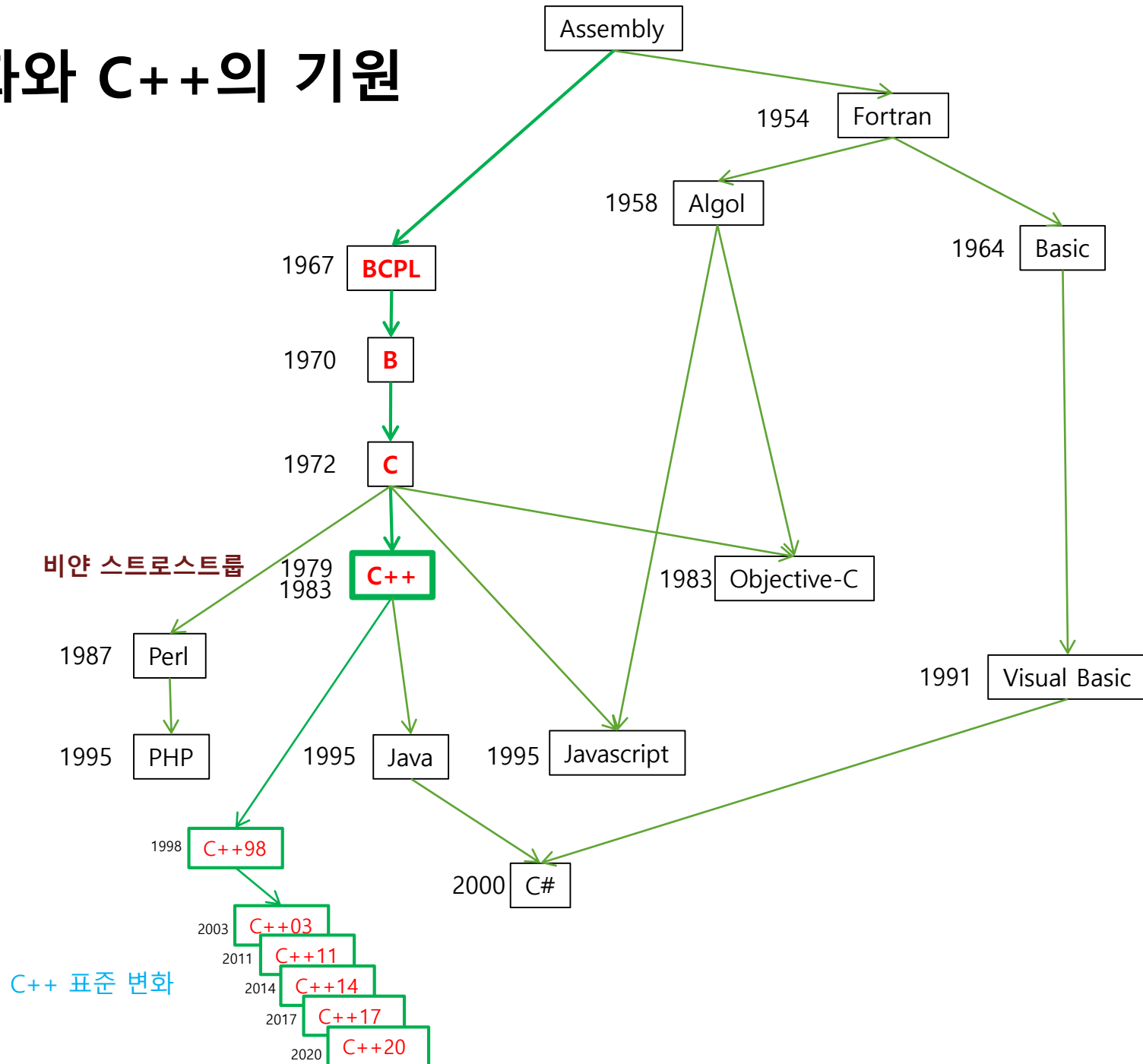


# C++ 개발환경

# 학습 목표

- 안녕하세요? 여러분!
- 이 강좌에서는 C++ 프로그래밍 언어를 사용하여 문제를 해결하는 방법을 배워가려고 합니다.
- 이 챕터는 "C++ 개요 및 개발 환경 설치"를 설명합니다.
- 이 챕터를 학습하시면 C++ 언어의 특징 및 프로그램 작성 방법을 알게 됩니다.
- 즐기는 학습이 되기를 바랍니다.

# 프로그래밍 언어의 진화와 C++의 기원



# C++ 언어의 주요한 설계 목적

- C 언어와의 호환성
  - C 언어의 문법 체계 계승
    - 소스 레벨 호환성 - 기존에 작성된 C 프로그램을 그대로 가져다 사용
    - 링크 레벨 호환성 - C 목적 파일과 라이브러리를 C++ 프로그램에서 링크
- 객체 지향 개념 도입
  - 캡슐화, 상속, 다형성
  - 소프트웨어의 재사용을 통해 생산성 향상
  - 복잡하고 큰 규모의 소프트웨어의 작성, 관리, 유지보수 용이
- 엄격한 타입 체크
  - 실행 시간 오류의 가능성을 줄임
  - 디버깅 편리
- 실행 시간의 효율성 저하 최소화
  - 실행 시간을 저하시키는 요소와 해결
  - 작은 크기의 멤버 함수 잦은 호출 가능성 -> 인라인 함수로 실행 시간 저하 해소

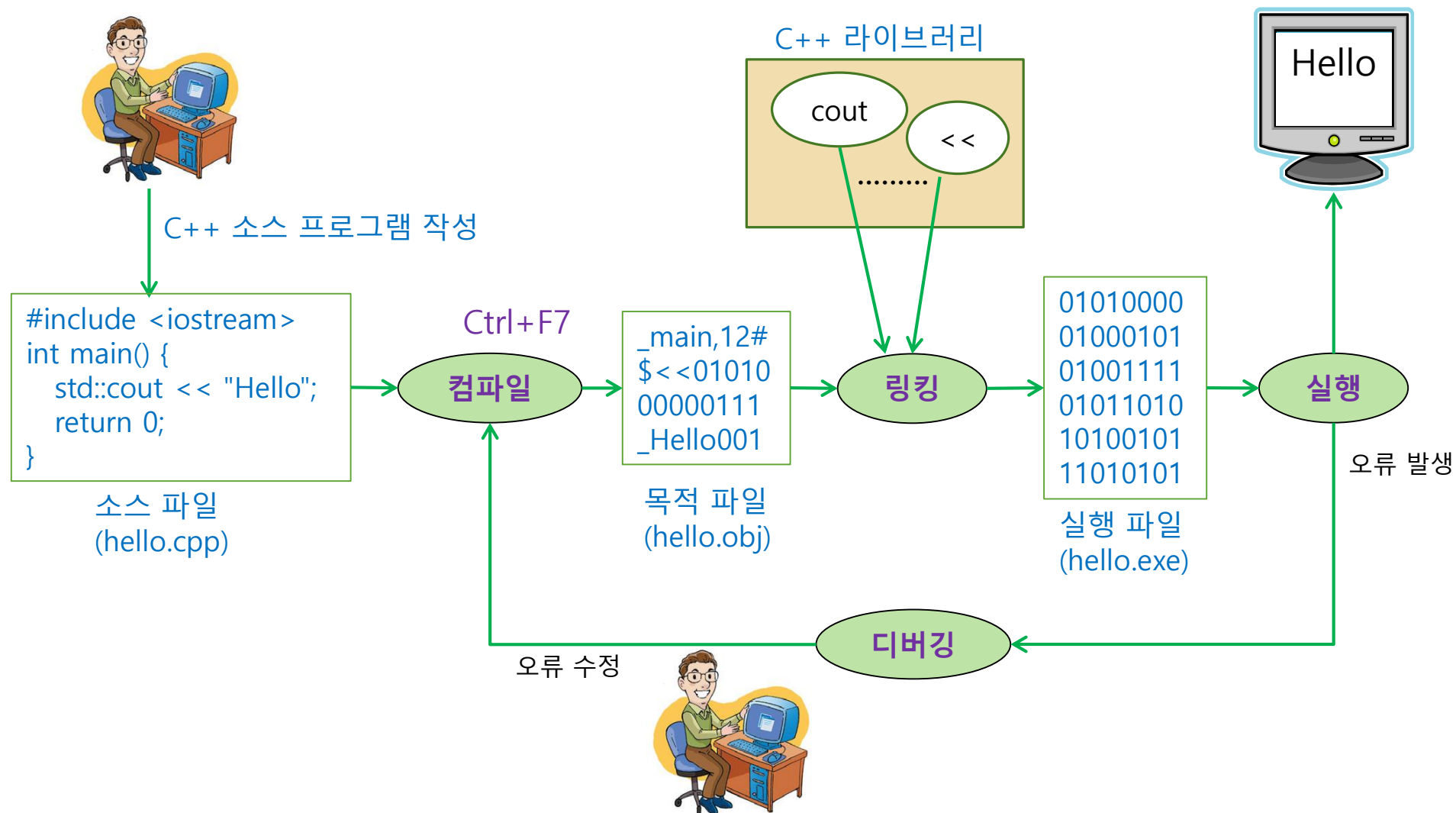
# C++ 언어에 추가한 기능

- 함수 중복(function overloading)
- 디폴트 매개 변수(default parameter)
- 참조와 참조 변수(reference)
- 참조에 의한 호출(call-by-reference)
- new/delete 연산자
- 연산자 재정의
- 제네릭 함수와 클래스
- 자동 타입 추론(auto)
- 범위 기반 루프
- 람다식
- 스마트 포인터
- STL 컨테이너(벡터, 맵 등)
- STL 알고리즘
- 예외

# C++ 언어의 아킬레스

- C++ 언어는 C 언어와의 호환성 추구
- 장점
  - 기존에 개발된 C 프로그램 코드 활용
- 단점
  - 캡슐화의 원칙이 무너지짐
    - C++에서 전역 변수와 전역 함수를 사용할 수 밖에 없음
    - 부작용(side effect) 발생 염려

# C++ 프로그램 개발 과정



# C++ 표준 라이브러리

- C++ 표준 라이브러리는 3 개의 그룹으로 구분
  - C 라이브러리
    - 기존 C 표준 라이브러리를 수용, C++에서 사용할 수 있게 한 함수들
    - 이름이 c로 시작하는 헤더 파일에 선언됨
  - C++ 입출력 라이브러리
    - 콘솔 및 파일 입출력을 위한 라이브러리
  - C++ STL 라이브러리
    - 제네릭 프로그래밍을 지원하기 위해 템플릿으로 작성된 라이브러리



# C++ 표준 라이브러리

algorithm	complex	exception	list	stack
bitset	csetjmp	fstream	locale	stdexcept
cassert	csignal	functional	map	strstream
cctype	cstdarg	iomanip	memory	streambuf
cerrno	cstddef	ios	new	string
cfloat	cstdio	iosfwd	numeric	typeinfo
ciso646	cstdlib	iostream	ostream	utility
climits	cstring	istream	queue	valarray
locale	ctime	iterator	set	vector
cmath	deque	limits	sstream	

\*`<new>` 헤더 파일은 STL에 포함되지 않는 기타 기능을 구현함

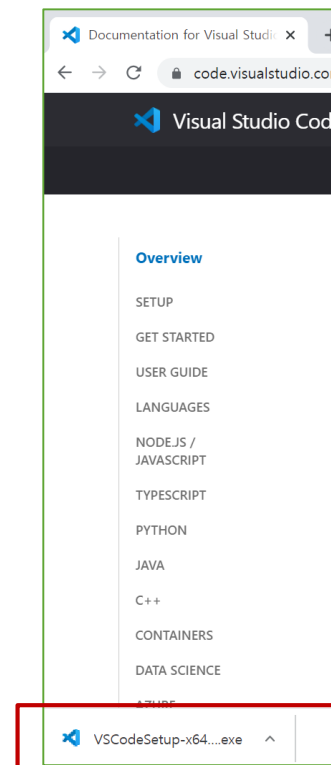
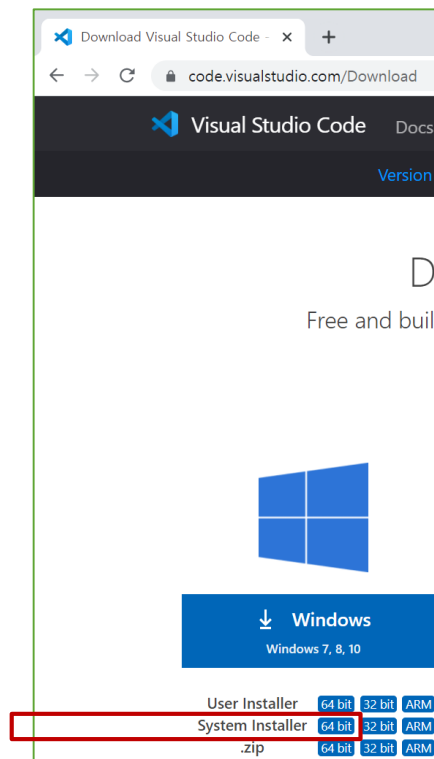
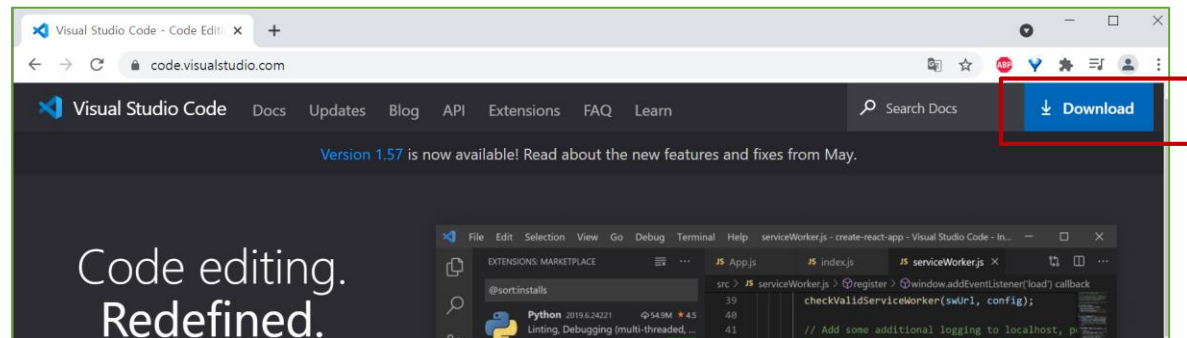
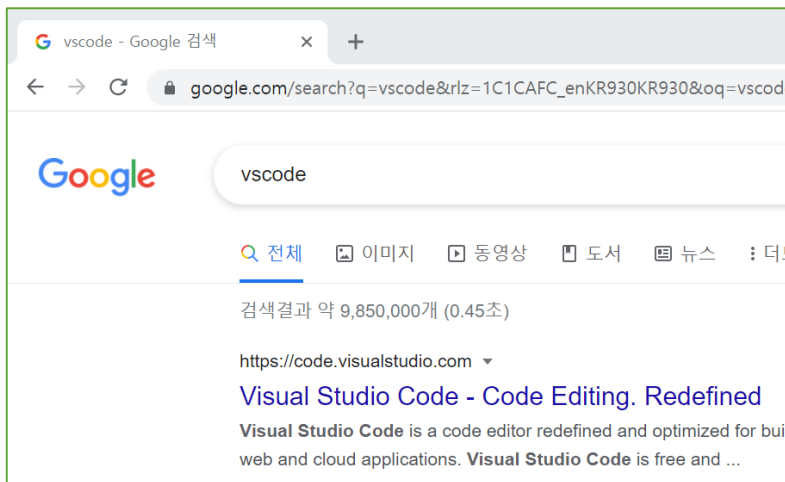
C++ 입출력 라이브러리

# 개발 환경 두 가지

- Microsoft Visual Studio Code
- Microsoft Visual Studio 20XX

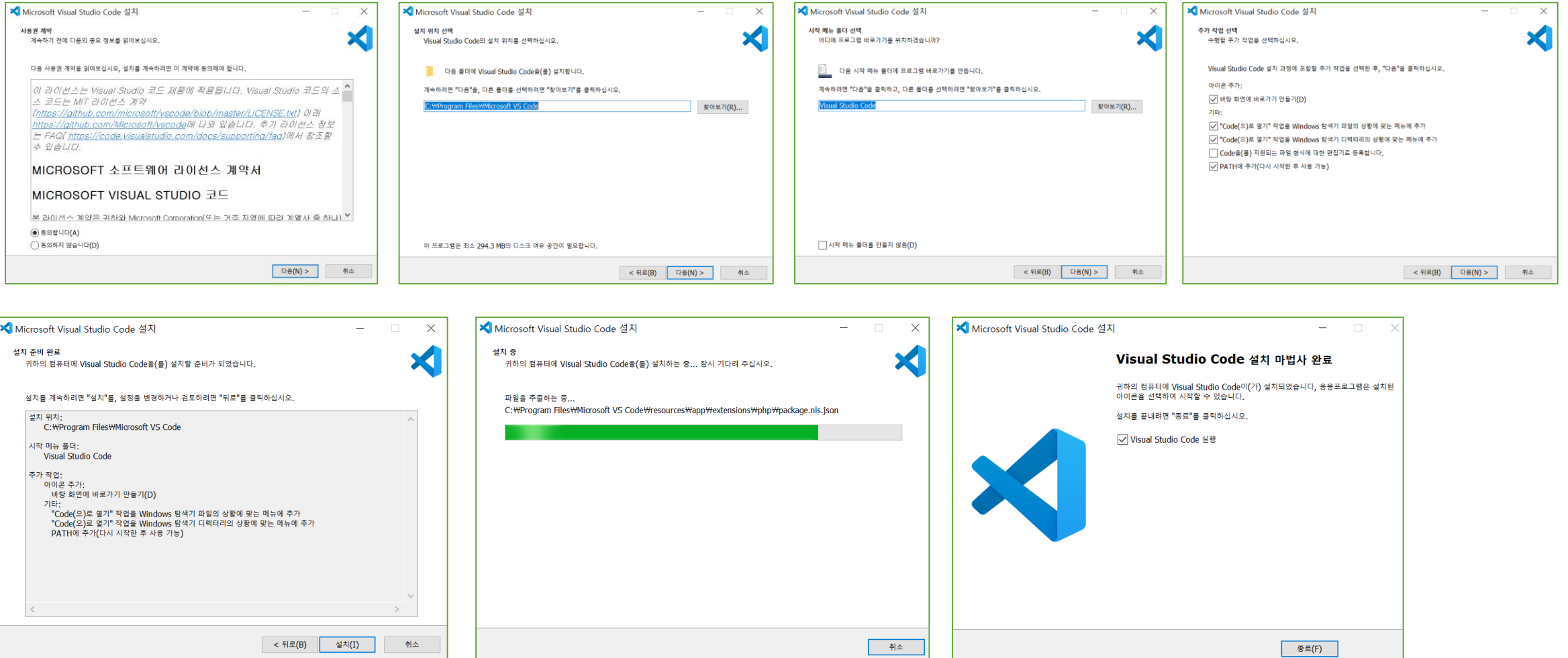
# VSCoDe 개발 환경 설치 및 실행 (1)

- VSCode 다운로드



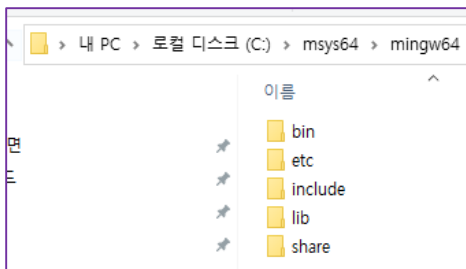
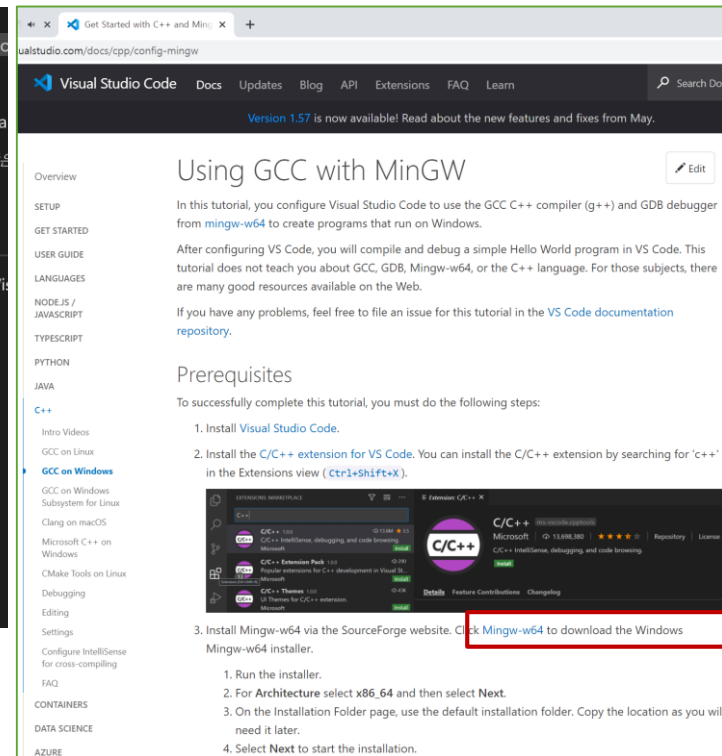
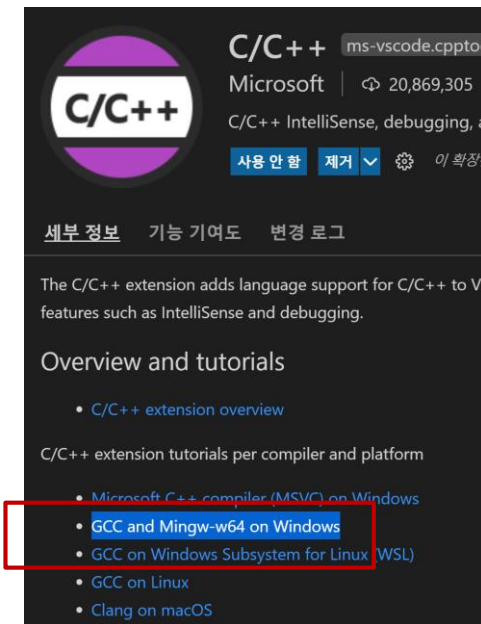
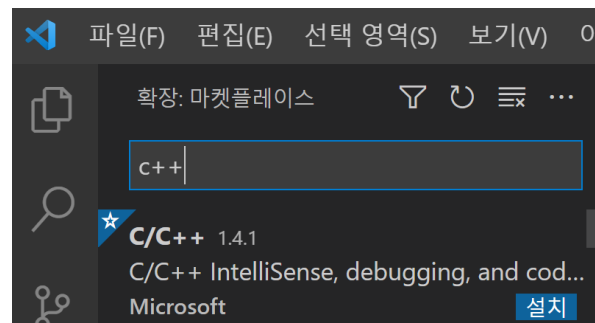
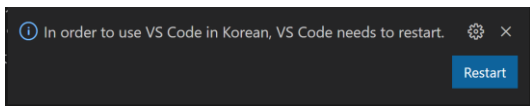
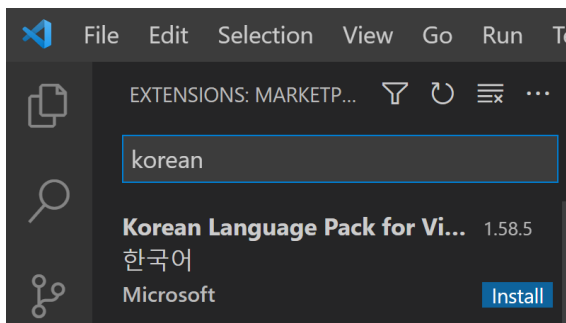
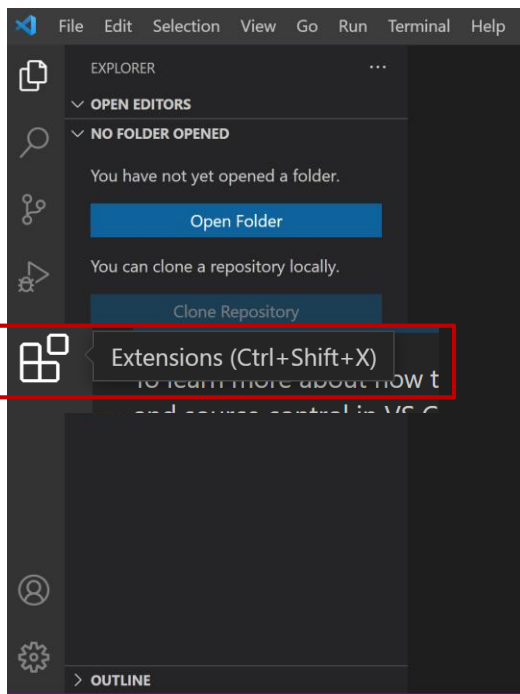
# VSCode 개발 환경 설치 및 실행 (2)

## • VSCode 설치



# VSCode 개발 환경 설치 및 실행 (3)

## Extension 설치



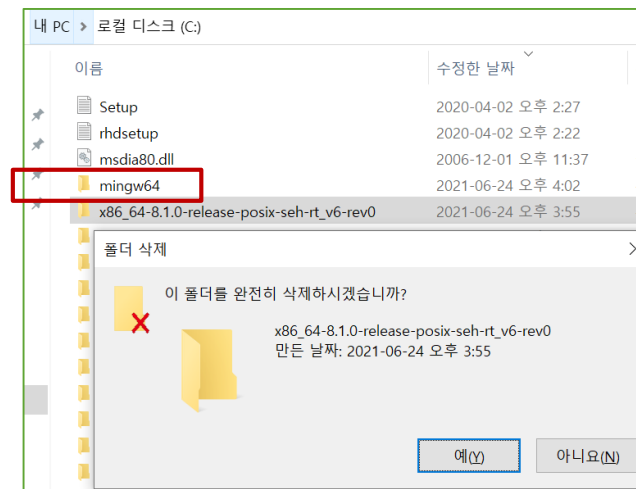
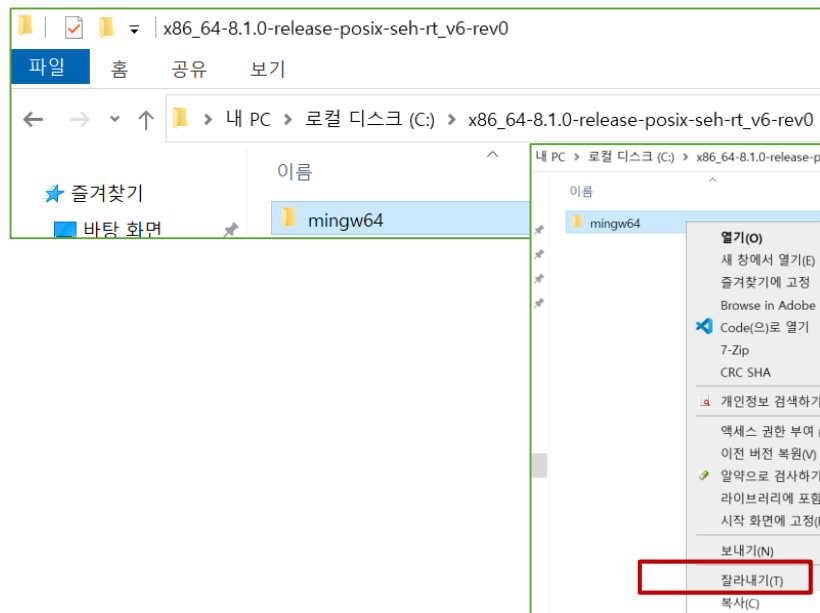
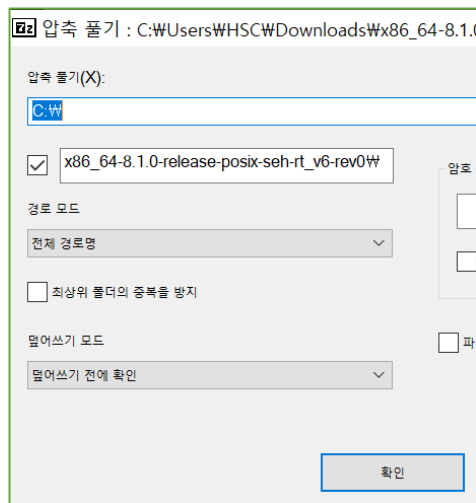
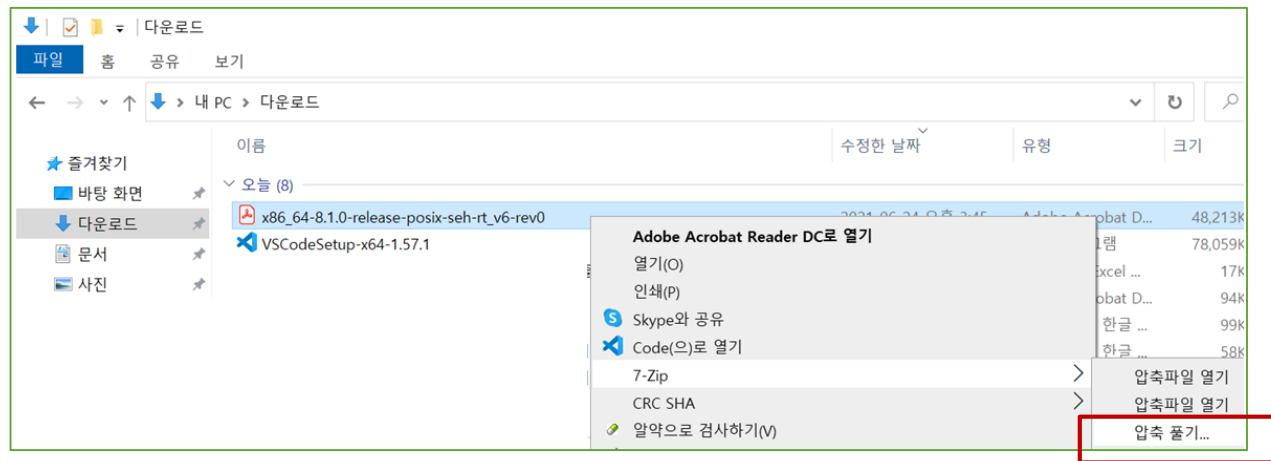
# VSCode 개발 환경 설치 및 실행 (4)

- MinGW-w64 설치(<https://sourceforge.net/>)

The first screenshot shows the SourceForge website with the search results for 'MinGW-w64'. The search results list 'MinGW - Minimalist GNU for Windows' and 'MinGW-w64 - for 32 and 64 bit Windows'. The second screenshot shows the 'Files' tab of the 'MinGW-w64' project, highlighting the 'Download Latest Version' button and the 'x86\_64-posix-seh' download link. The third screenshot shows the download progress bar for the selected file, 'x86\_64-8.1.0-release-posix-seh-rt\_v6-rev0.7z'.

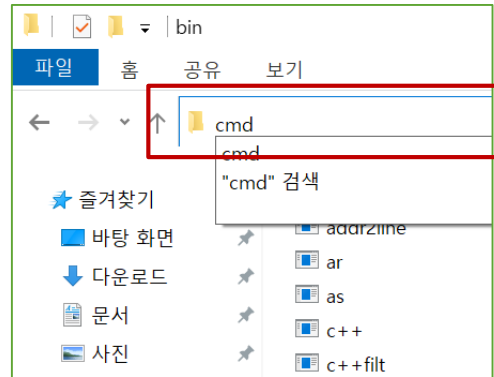
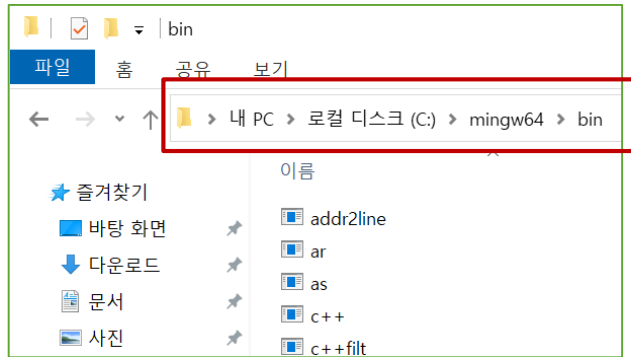
# VSCode 개발 환경 설치 및 실행 (5)

- MinGW-w64 디렉토리 설정



# VSCode 개발 환경 설치 및 실행 (6)

- MinGW-w64 설치 확인



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1052]
(c) Microsoft Corporation. All rights reserved.

C:\mingw64\bin>gcc --version
gcc (x86_64-posix-seh-rev0, Built by MinGW-W64 project)
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying
warranty; not even for MERCHANTABILITY or FITNESS
for a particular purpose.

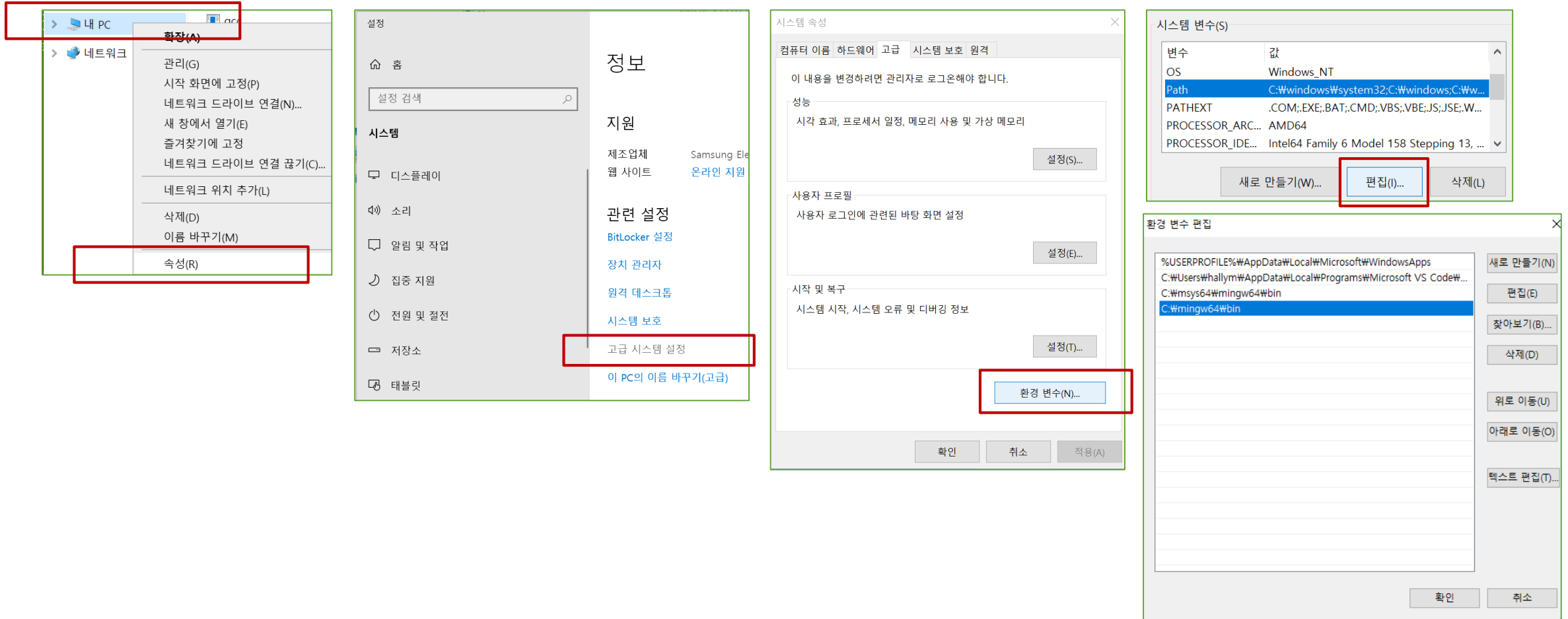
C:\mingw64\bin>g++ --version
g++ (x86_64-posix-seh-rev0, Built by MinGW-W64 project)
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying
warranty; not even for MERCHANTABILITY or FITNESS
for a particular purpose.

C:\mingw64\bin>
```



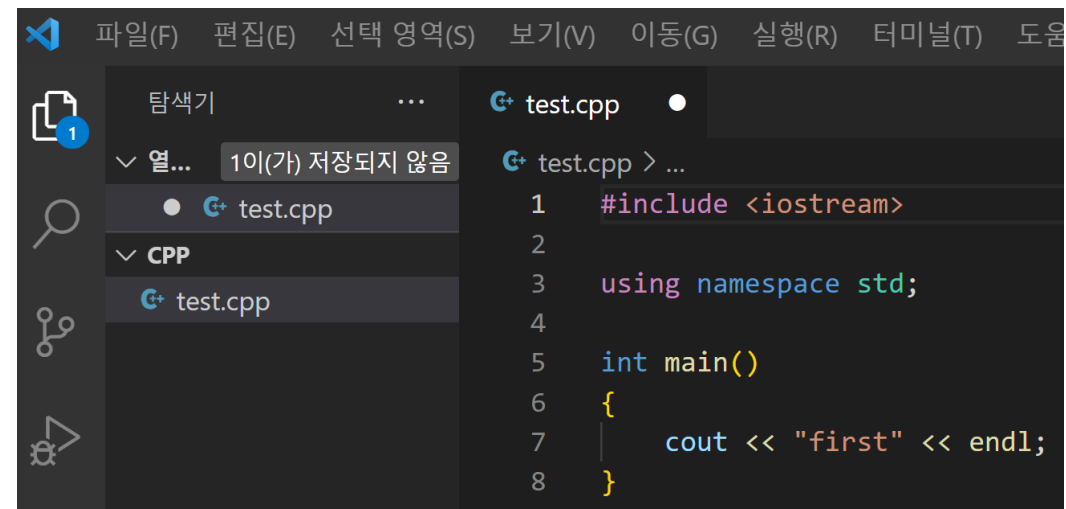
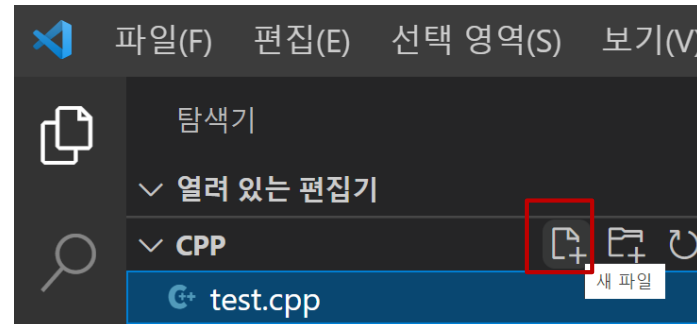
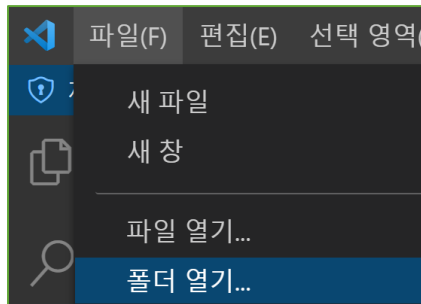
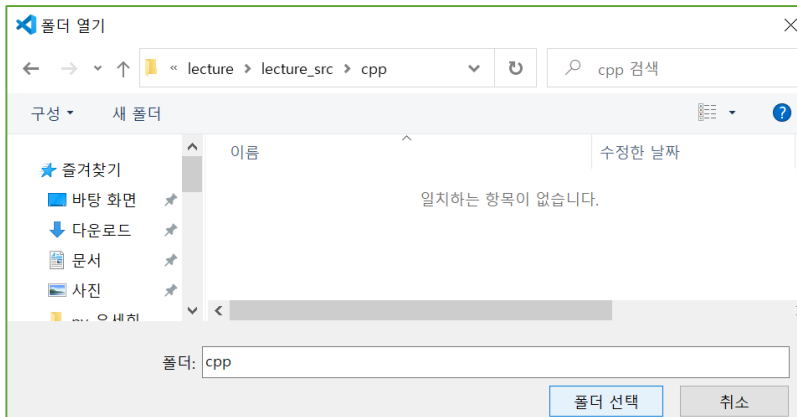
# VSCode 개발 환경 설치 및 실행 (7)

## • 환경변수 설정



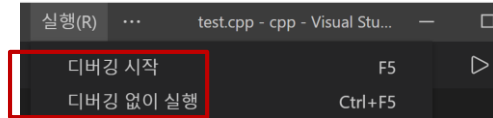
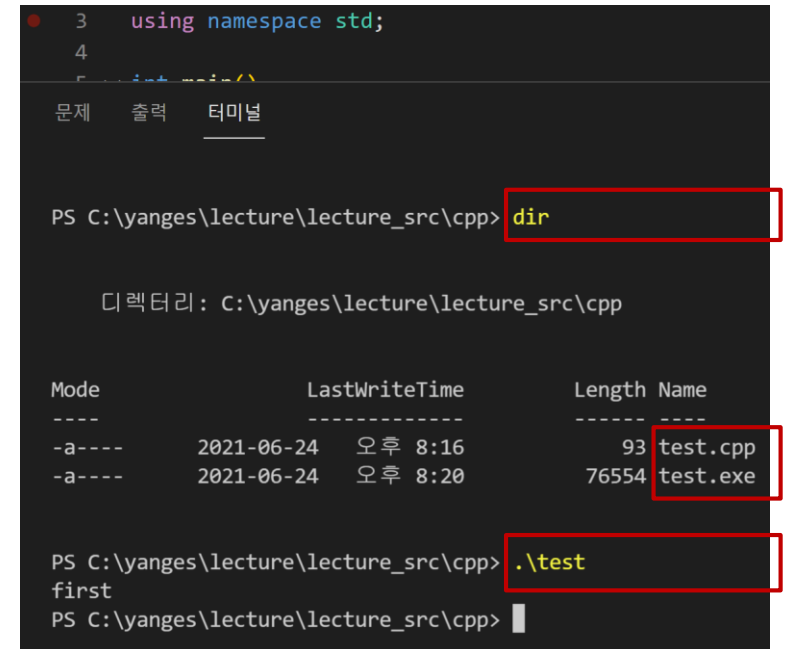
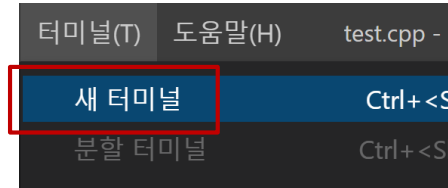
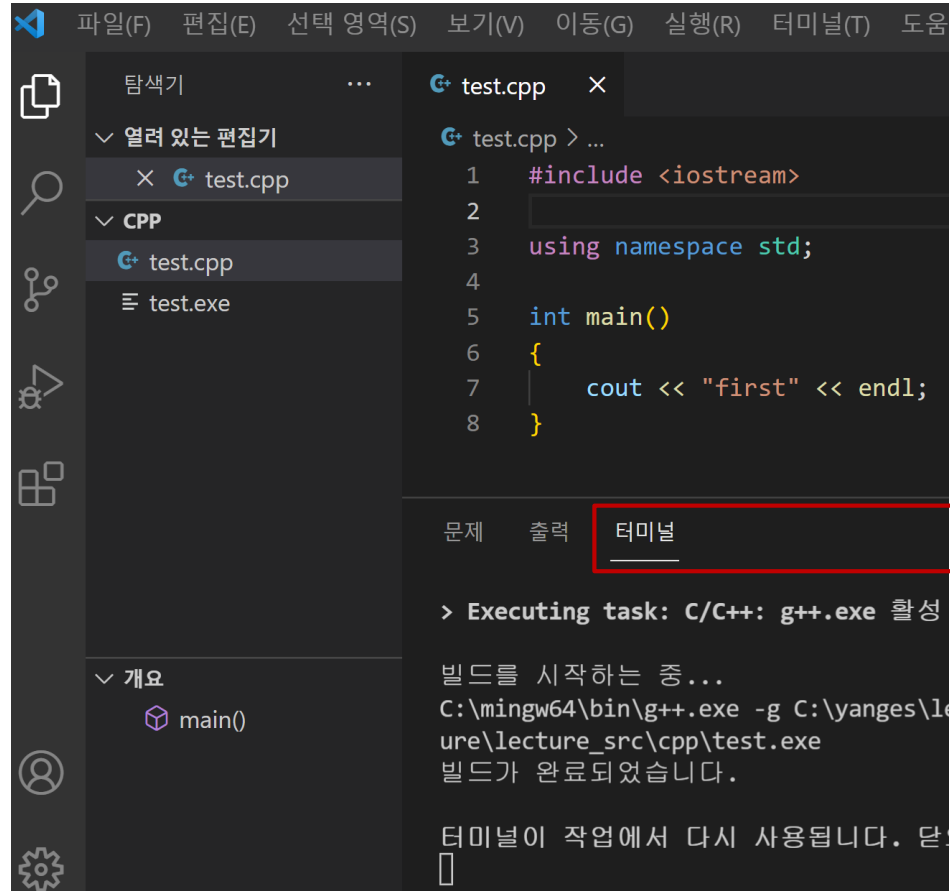
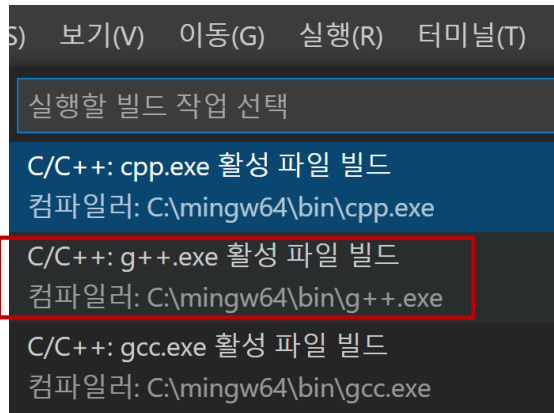
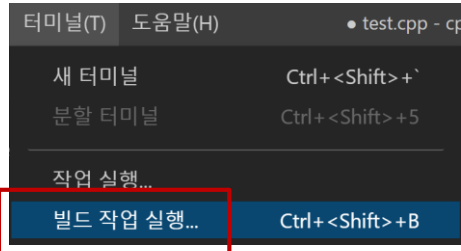
# VSCode 개발 환경 설치 및 실행 (8)

- 작업 폴더 생성 / cpp 파일 생성




# VSCode 개발 환경 설치 및 실행 (9)

## • 빌드 및 실행

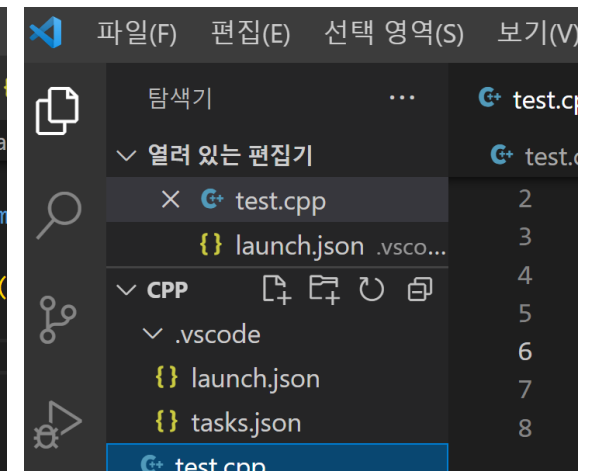
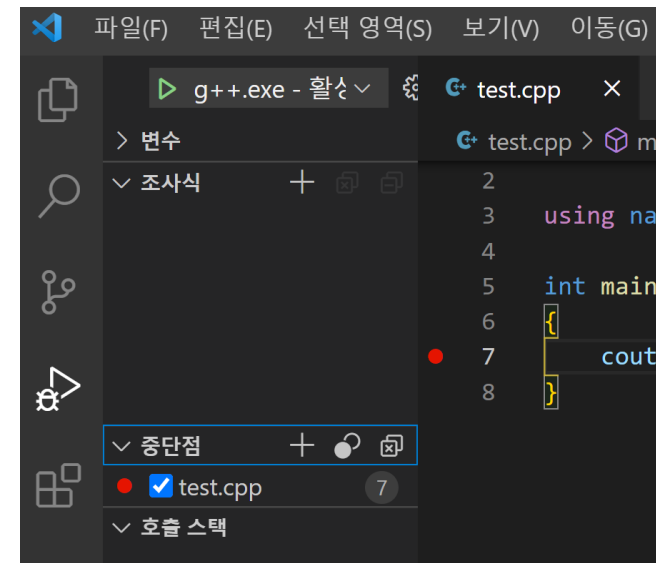
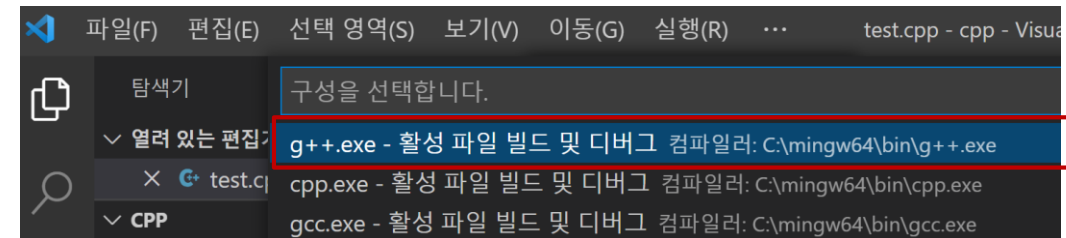
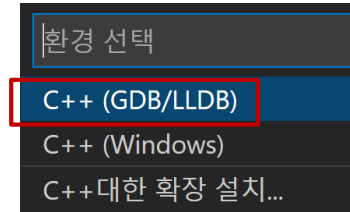
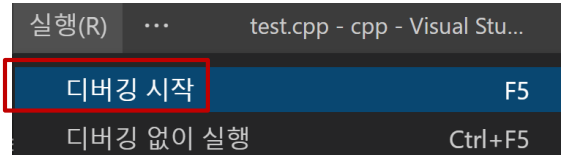


# VSCode 개발 환경 설치 및 실행 (10)

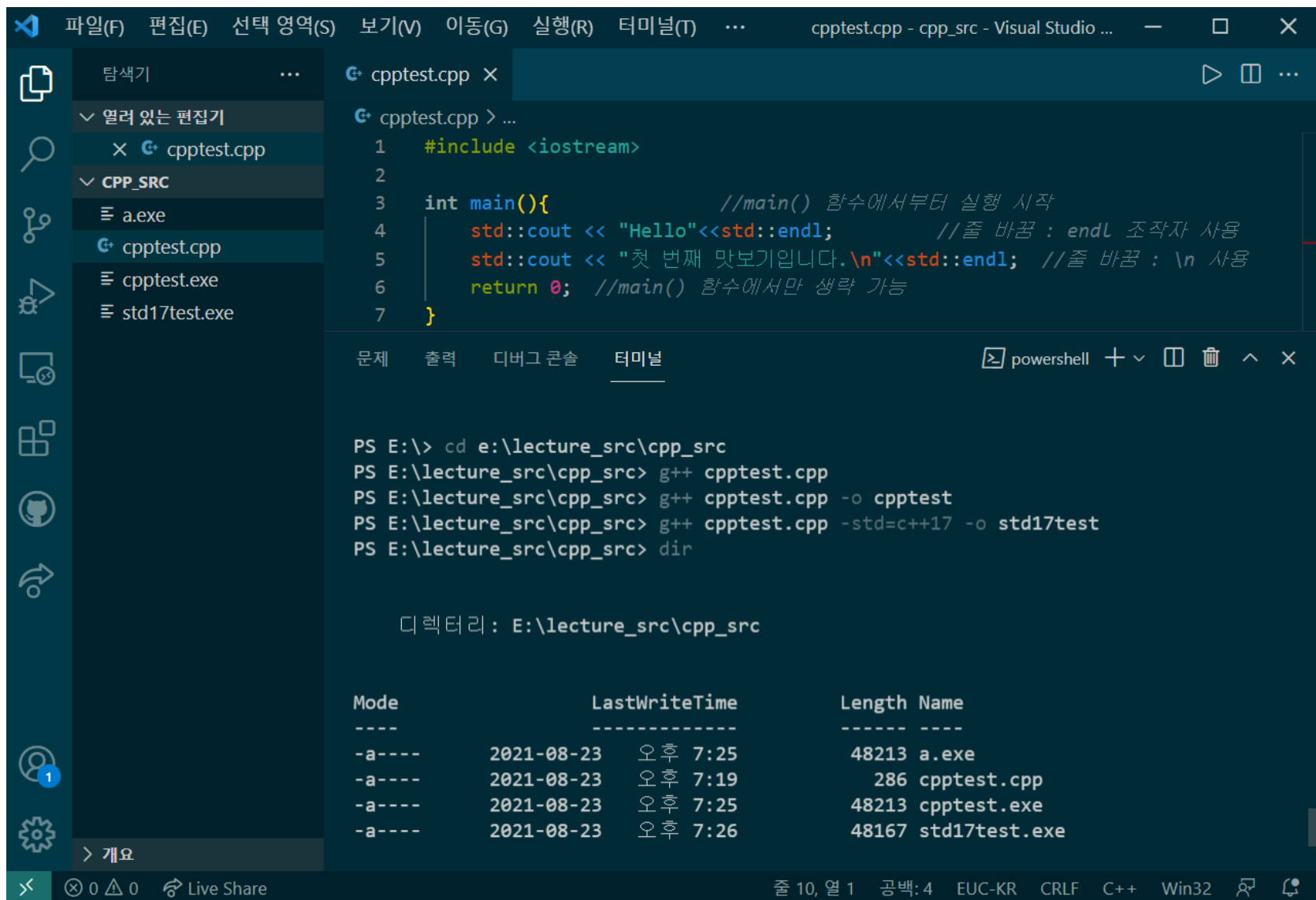
- Breakpoint 설정



```
test.cpp > ...
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7  cout << "first" << endl;
8  }
```

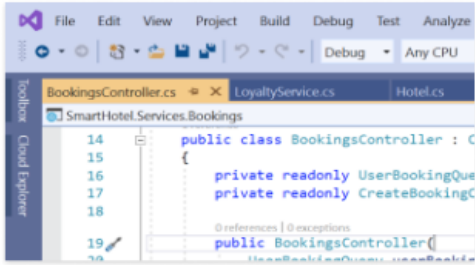


# VSCode 개발 환경 설치 및 실행 (11)



# Microsoft Visual Studio 20XX 설치

- 아래의 사이트에서 프로그램을 다운로드하여 설치
  - <https://visualstudio.microsoft.com/ko/>



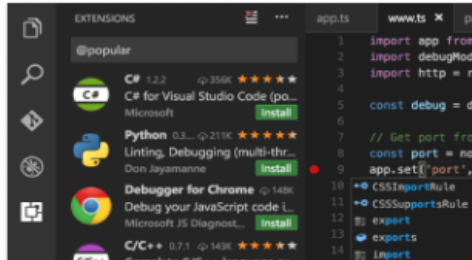
코딩, 디버그, 테스트 및 모든 플랫폼에 배포할 수 있는 완전한 기능을 갖춘 IDE

Visual Studio 다운로드

Community 2019

Professional 2019

Enterprise 2019

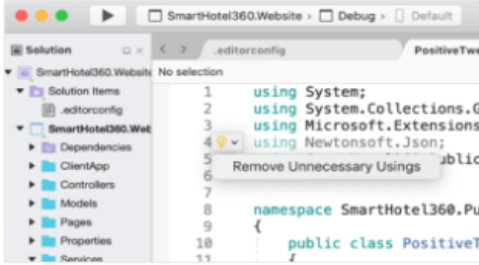


모든 OS 편집 및 디버깅

Visual Studio Code를 사용하면 다음에 동의하는 것입니다.  
[라이선스 및 개인정보처리방침](#)

Download Visual Studio Code

[자세히 보기 >](#)



.NET을 사용하여 iOS, Android 및 웹용 앱 및 게임 개발

[라이선스 활성화에 대한 자세한 정보](#)

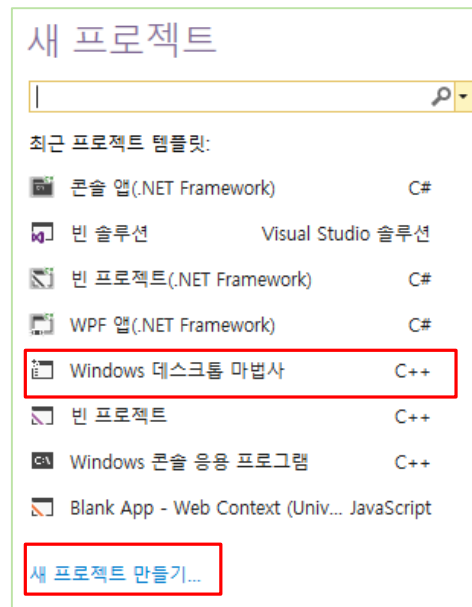
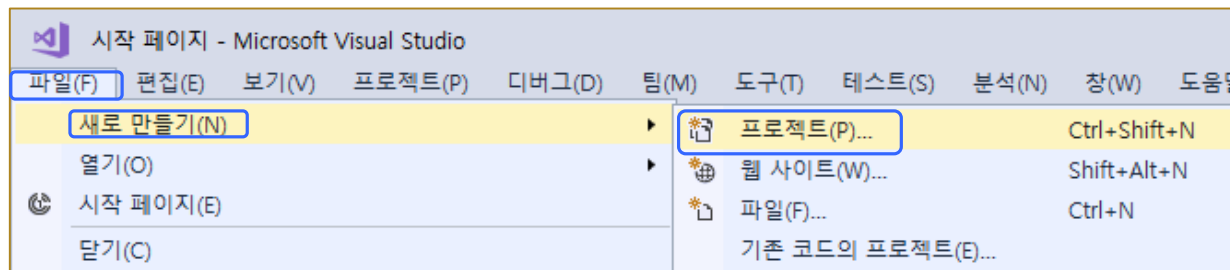
VISUAL STUDIO FOR MAC 다운로드

[자세히 보기 >](#)

- 이전 버전이 필요하면 <https://visualstudio.microsoft.com/ko/vs/older-downloads/> 에서 다운로드

# Visual Studio 20XX 프로그램 작성 및 실행(1)

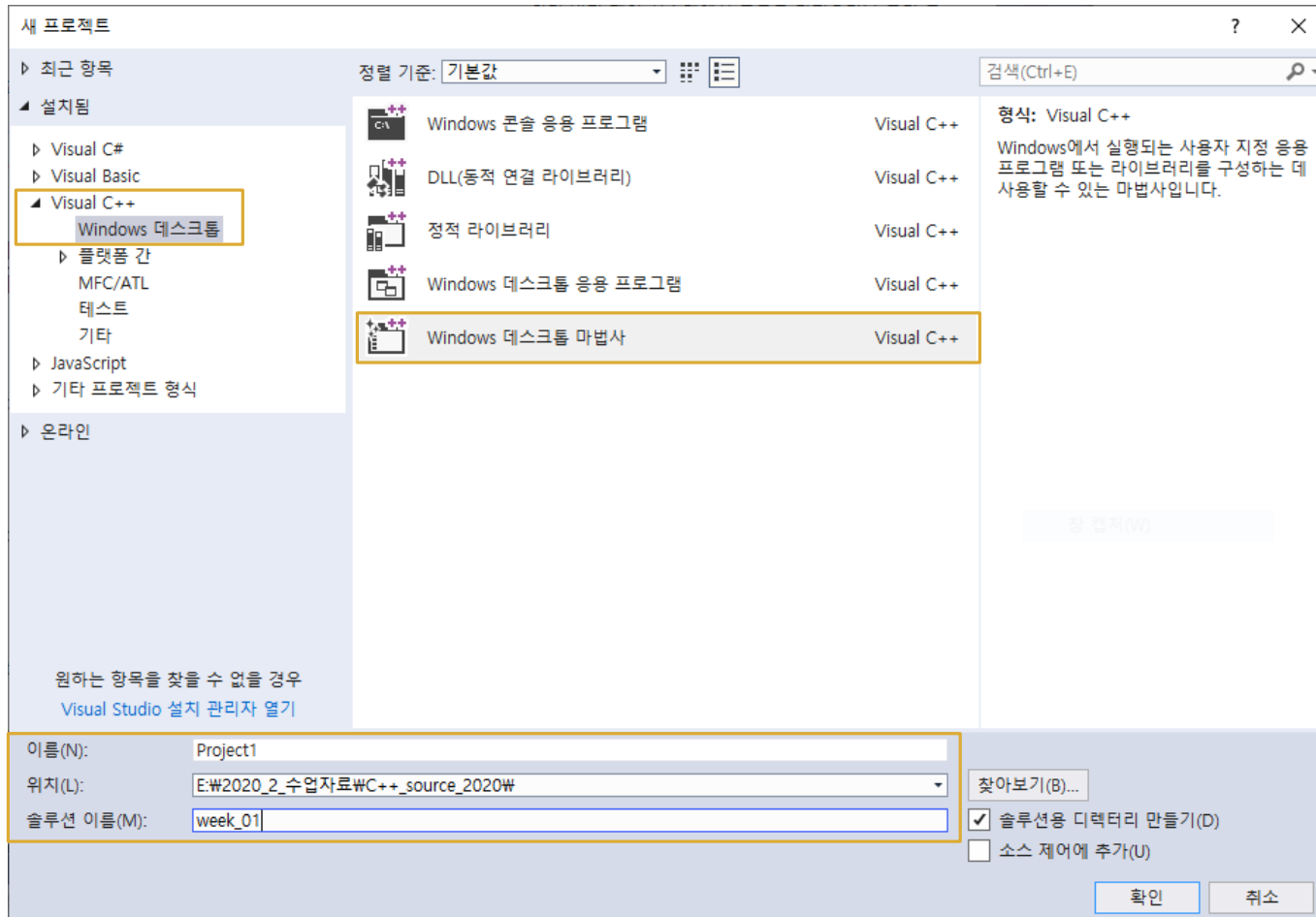
1. 설치한 Visual Studio 20XX 실행
2. 프로젝트 만들기
  - 시작 페이지 -> '새 프로젝트 만들기'
  - 파일(F)->새로 만들기(N)->프로젝트(P)



- 솔루션과 프로젝트
  - 프로젝트: 여러 소스와 관련 파일을 저장 · 관리하는 단위로, 프로젝트 이름으로 하나의 실행파일을 생성
  - 솔루션: 하나 이상의 프로젝트를 저장 · 관리하는 단위

# Visual Studio 20XX 프로그램 작성 및 실행(2)

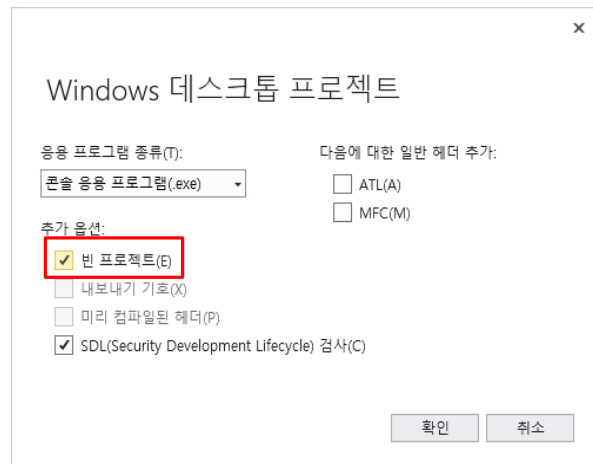
3. 새 프로젝트 창 -> Visual C++ -> Windows 데스크톱 -> Windows 데스크톱 마법사 선택
4. 솔루션 이름과 프로젝트 이름, 저장 위치를 지정한다





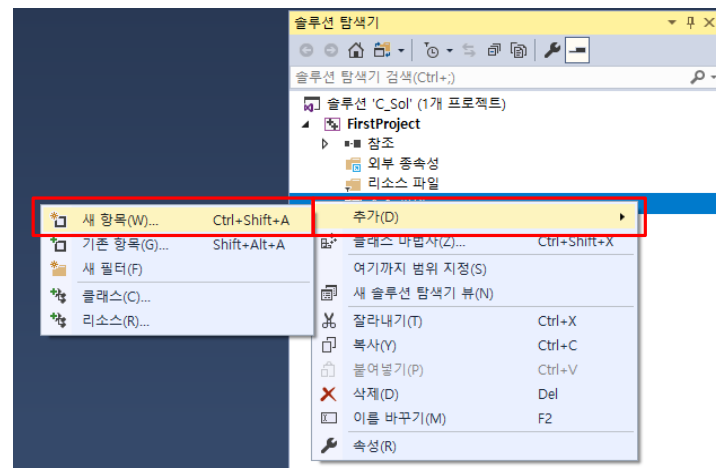
# Visual Studio 20XX 프로그램 작성 및 실행(3)

## 5. Windows 데스크톱 프로젝트-> 빈 프로젝트 체크



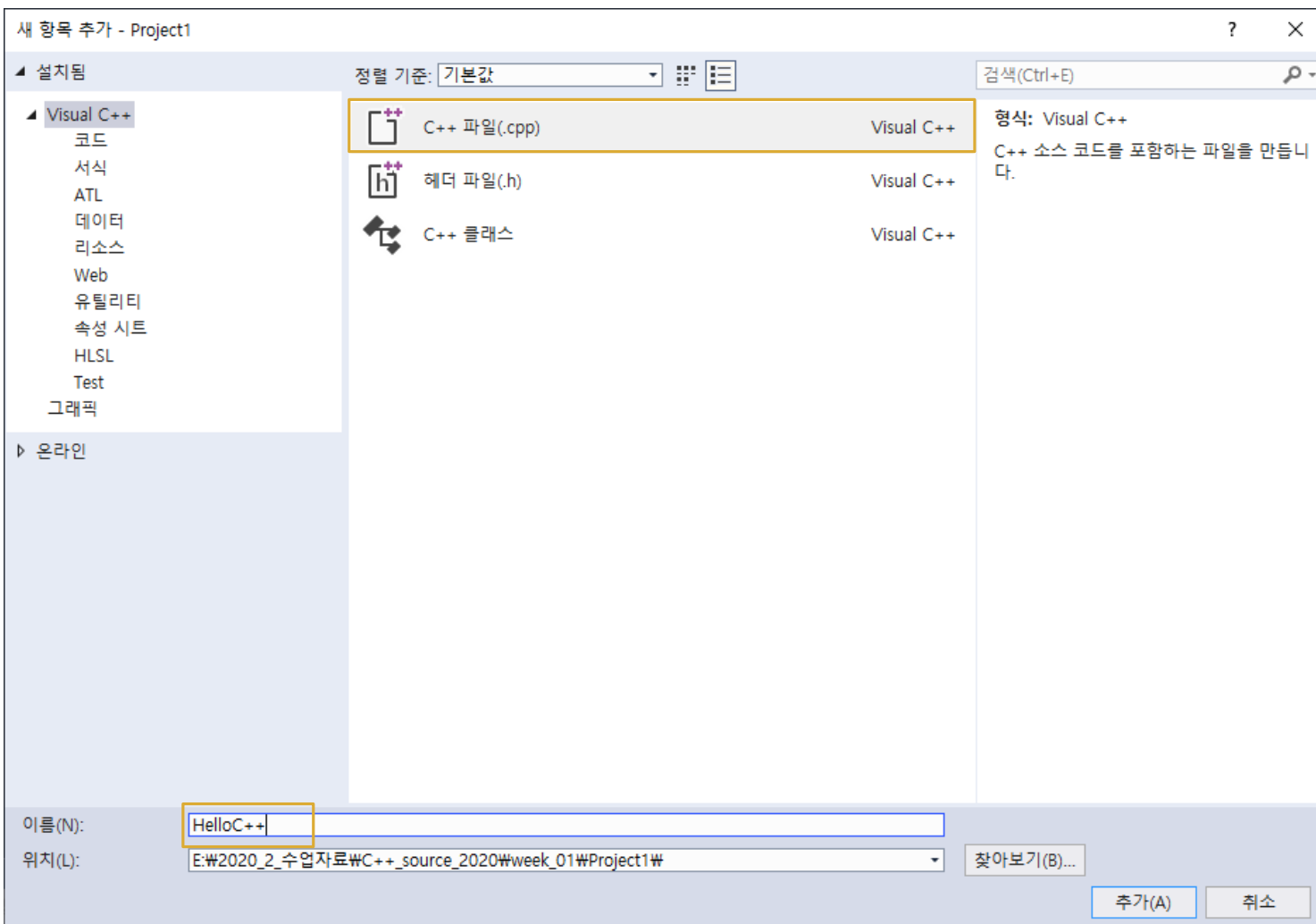
## 6. 솔루션 탐색기

-> 소스 파일(마우스 오른쪽 버튼 클릭) -> 추가 -> 새 항목



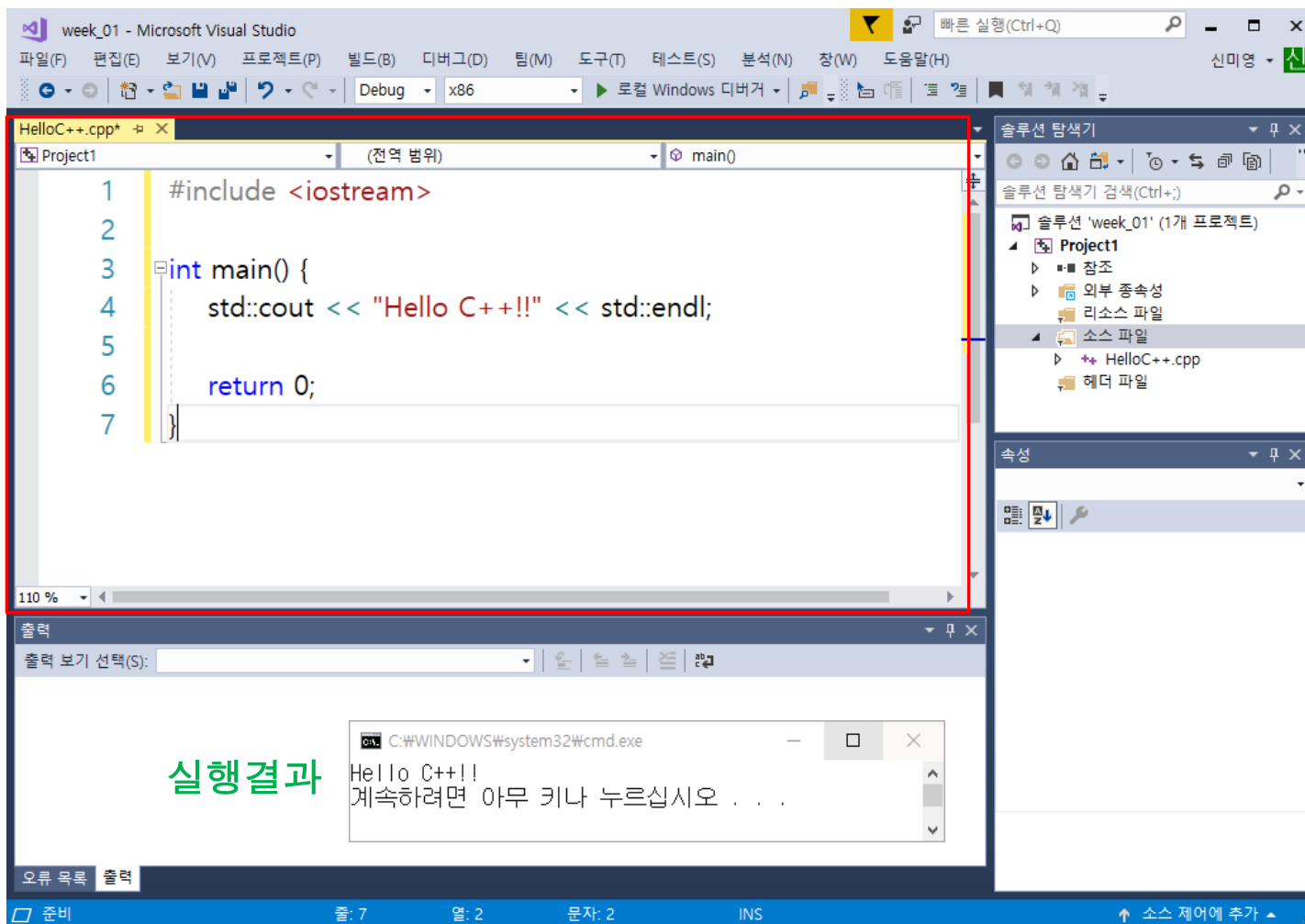
# Visual Studio 20XX 프로그램 작성 및 실행(4)

7. 새 항목 추가 -> C++ 파일(.cpp) -> 파일명 입력, 확장자 생략 시 자동으로 cpp



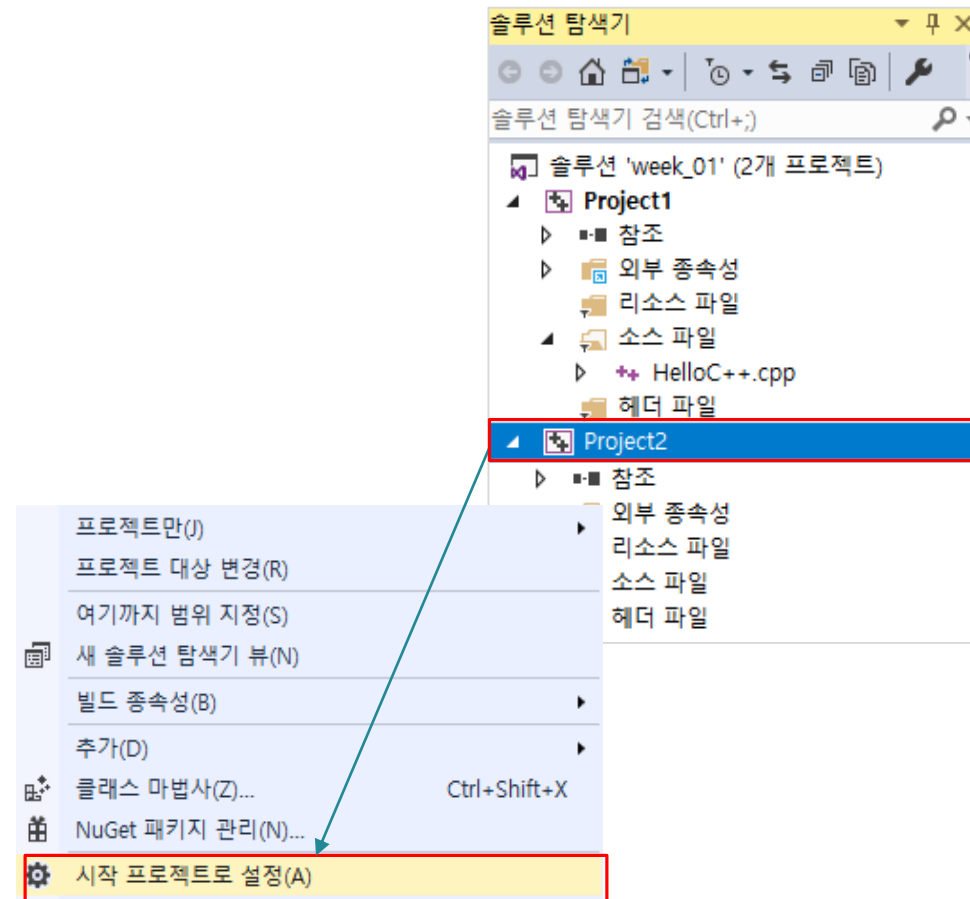
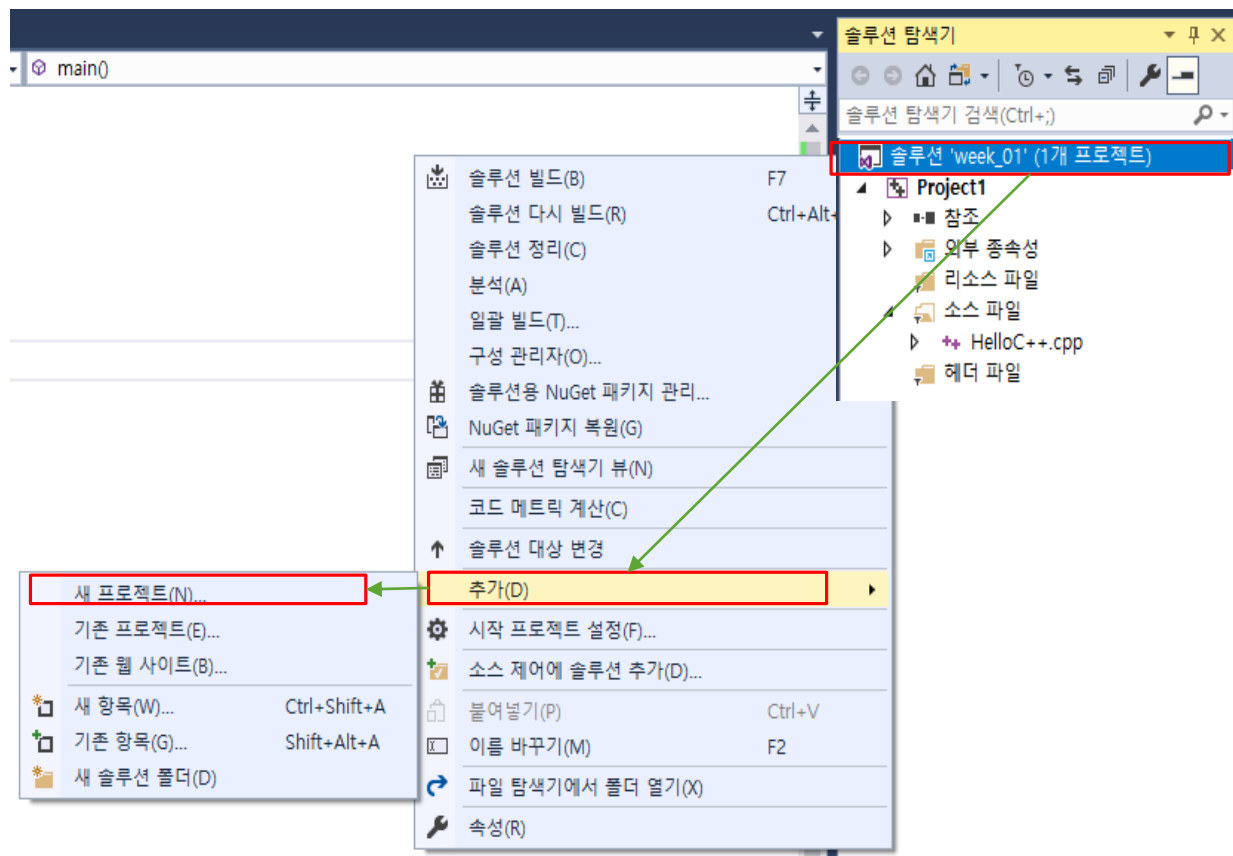
# Visual Studio 20XX 프로그램 작성 및 실행(5)

## • 프로그램 작성



# Visual Studio 20XX 프로그램 작성 및 실행(6)

- 솔루션에 새 프로젝트 추가하기



# Q & A

- “C++ 개발 환경”에 대한 학습이 모두 끝났습니다.
- 모든 내용을 이해 하셨나요?
- 아직 이해가 안되는 내용이 있다면 다시 한번 복습하시기 바랍니다.
- 질문은 한림 SmartLEAD 쪽지 또는 e-mail 또는 전화상담을 이용하시기 바랍니다.



- 과제가 출제되었습니다. 마감시간에 늦지 않도록 주의해 주세요.
- 수고하셨습니다.^^