

Windows, MacOS에서 OpenCV 설치법

2020년 1월

부록0: Visual Studio 2019 설치

Visual Studio 2019 다운로드

- <https://visualstudio.microsoft.com/ko/vs/>

The screenshot shows the Visual Studio 2019 download page. The header includes the Microsoft logo, 'Visual Studio' navigation links, and a '무료 Visual Studio' button. The main heading is 'Visual Studio 2019' with the tagline '더 신속한 코딩. 보다 스마트한 작업. 동급 최고의 IDE를 사용하여 미래를 만들어 보세요.' Below this, a red box highlights the 'Visual Studio 다운로드' dropdown menu, which lists 'Community 2019', 'Professional 2019', and 'Enterprise 2019'. A tooltip for 'Community 2019' is visible, stating '개별 개발자, 학습 용도 및 오픈 소스는 무료로 제공됩니다.' The bottom of the image shows a preview of the Visual Studio IDE interface.

Microsoft | Visual Studio Visual Studio 2019 기능 ▾ 버전 ▾ 다운로드 지원 ▾ 구독자 액세스 무료 Visual Studio Microsoft 전체 ▾ 검색 🔍 Sign in 👤

Visual Studio 2019

더 신속한 코딩. 보다 스마트한 작업. 동급 최고의 IDE를 사용하여 미래를 만들어 보세요.

Visual Studio 다운로드 ▾

- Community 2019 ↓
- Professional 2019 ↓
- Enterprise 2019 ↓

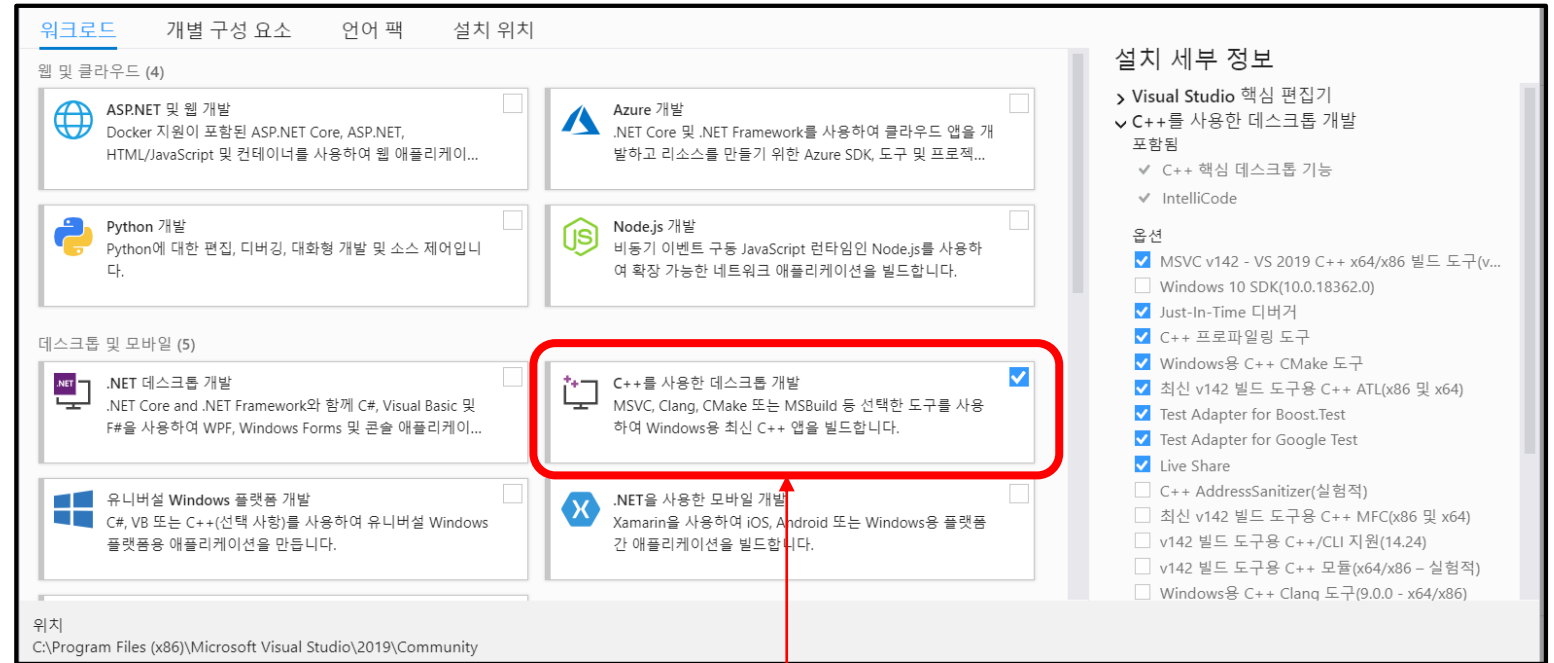
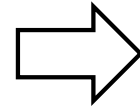
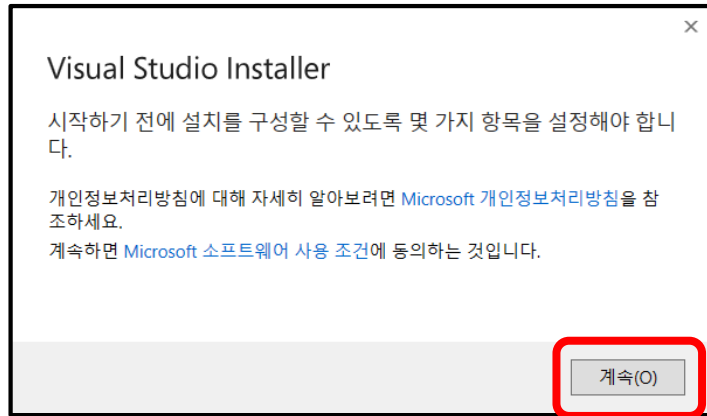
개별 개발자, 학습 용도 및 오픈 소스는 무료로 제공됩니다.

ConsoleApp1 - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Quick Launch (Ctrl+Q) 🔍 Sign in

Visual Studio 2019 다운로드



설치

부록 1-1: Windows에서 OpenCV 사용하기

(환경 변수 설정 없이 opencv 환경 설정 version)

웹에서 OpenCV 다운로드하기

- www.opencv.org

[About](#)[Releases](#)[Courses](#)[Resources](#)[Partnership](#)[News](#)[License](#)

OpenCV @CES 2020

See you at our private party on Jan 7

[Learn More](#)

웹에서 OpenCV 다운로드하기

Releases

* OpenCV – 4.2.0

📅 2019-12-23

📄 Docs

📁 Sources

🐙 GitHub

🪟 Windows

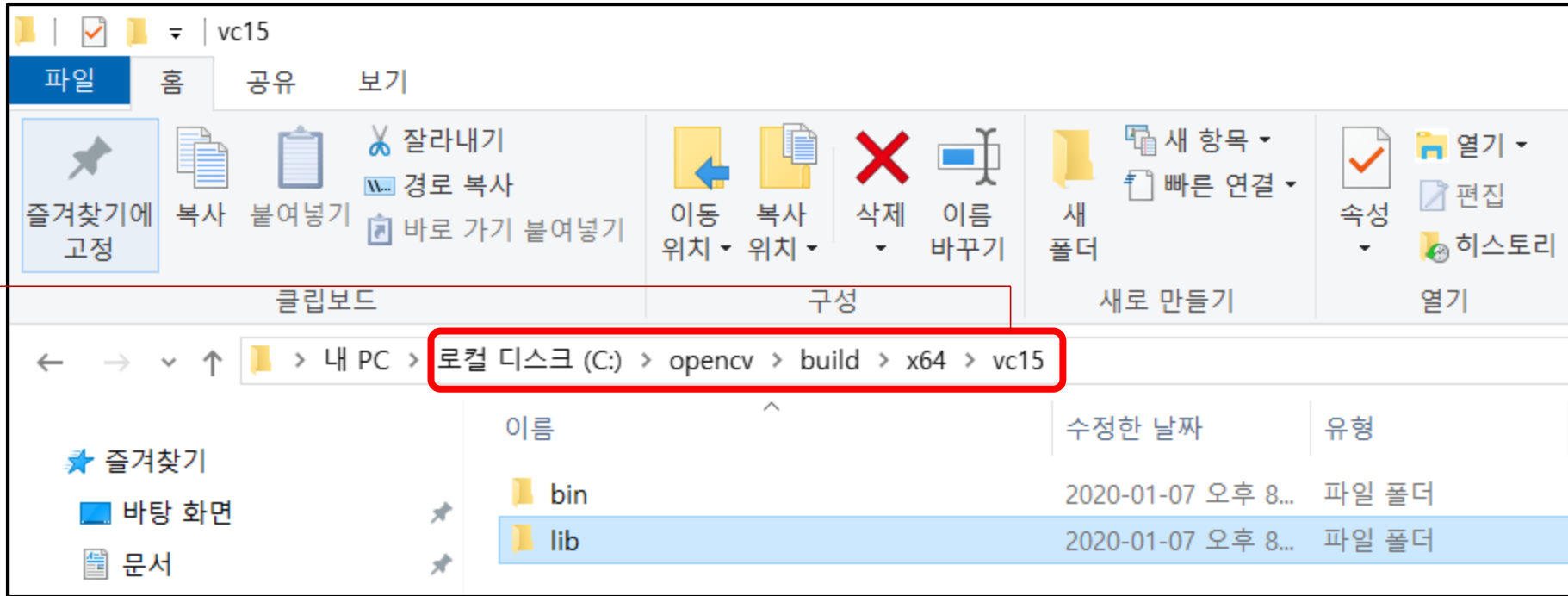
🍏 iOS pack

🤖 Android

📄 Release Notes

OpenCV 사용 환경 설정

: OpenCV 다운로드한 위치 확인



(opencv 설치 위치가 C:\wopencv이고, 64-bit 윈도우에서 Visual Studio 2017 또는 Visual Studio 2019를 쓴다고 가정)

→ *주의 **vc15**가 **Visual Studio 2017 또는 Visual Studio 2019**에 해당하는 컴파일러 버전이다.
vc14는 **Visual Studio 2015** 컴파일러 버전이다.

Visual Studio (VS) 설정하기

- Configuration 바꾸어 x64 플랫폼으로 설정하기
- openCV 라이브러리 다운로드 받기
- openCV 라이브러리 및 헤더파일 경로 설정하기

→ 다음 슬라이드를 보면서 설정하세요!

Visual Studio : New project 생성



Visual Studio : New project 생성

새 프로젝트 구성

Windows 데스크톱 마법사 C++ Windows 데스크톱 콘솔 라이브러리

프로젝트 이름(N)
opencv_test

위치(L)
C:\Users\Wjeonghoon\source\repos

솔루션 이름(M)
opencv_test

☐ 솔루션 및 프로젝트를 같은 디렉터리

Windows 데스크톱 프로젝트

애플리케이션 종류(T)
콘솔 애플리케이션(.exe)

추가 옵션:
☒ 빈 프로젝트(E)
☐ 미리 컴파일된 헤더(P)
☐ 기호 내보내기(X)
☐ MFC 헤더(M)

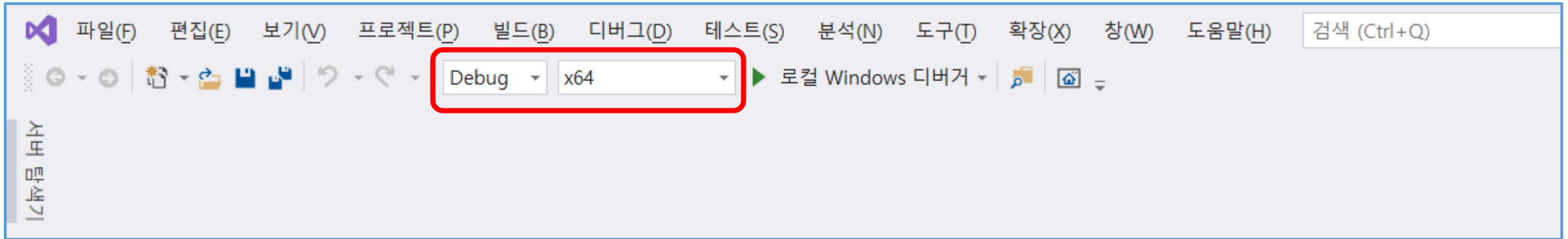
팁: 빈 프로젝트 템플릿을 사용하여 이 종류의 프로젝트를 만들 수도 있습니다.

확인 취소

뒤로(B) 만들기(C)

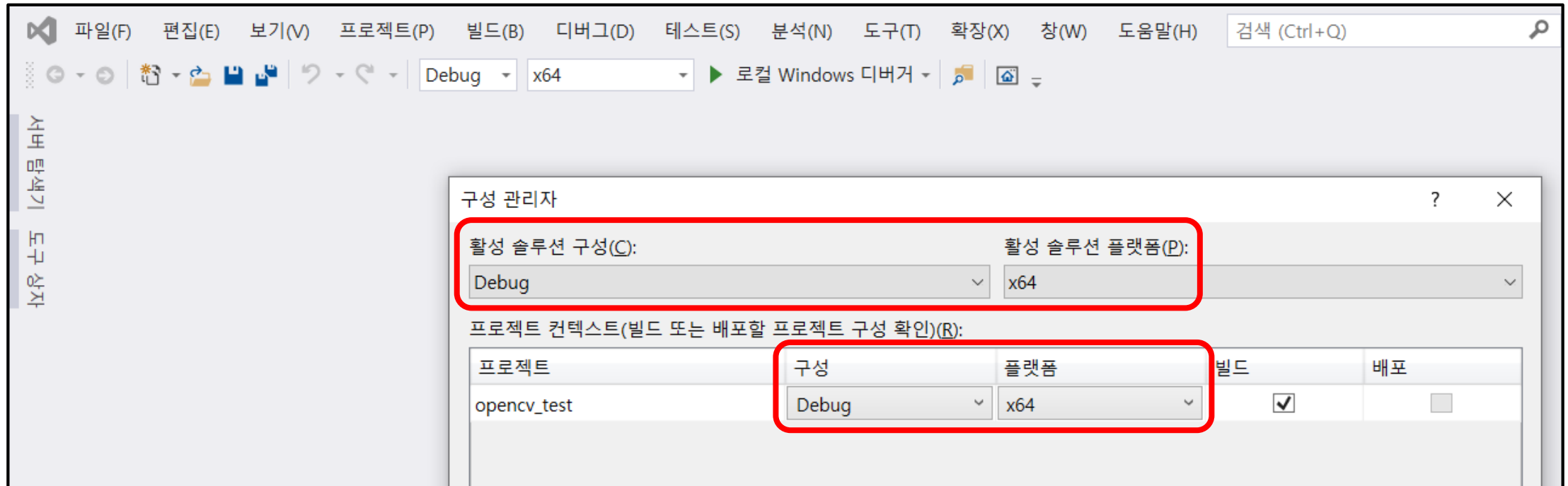
Visual Studio 설정하기

- Configuration 바꾸어 x64 플랫폼으로 설정하기

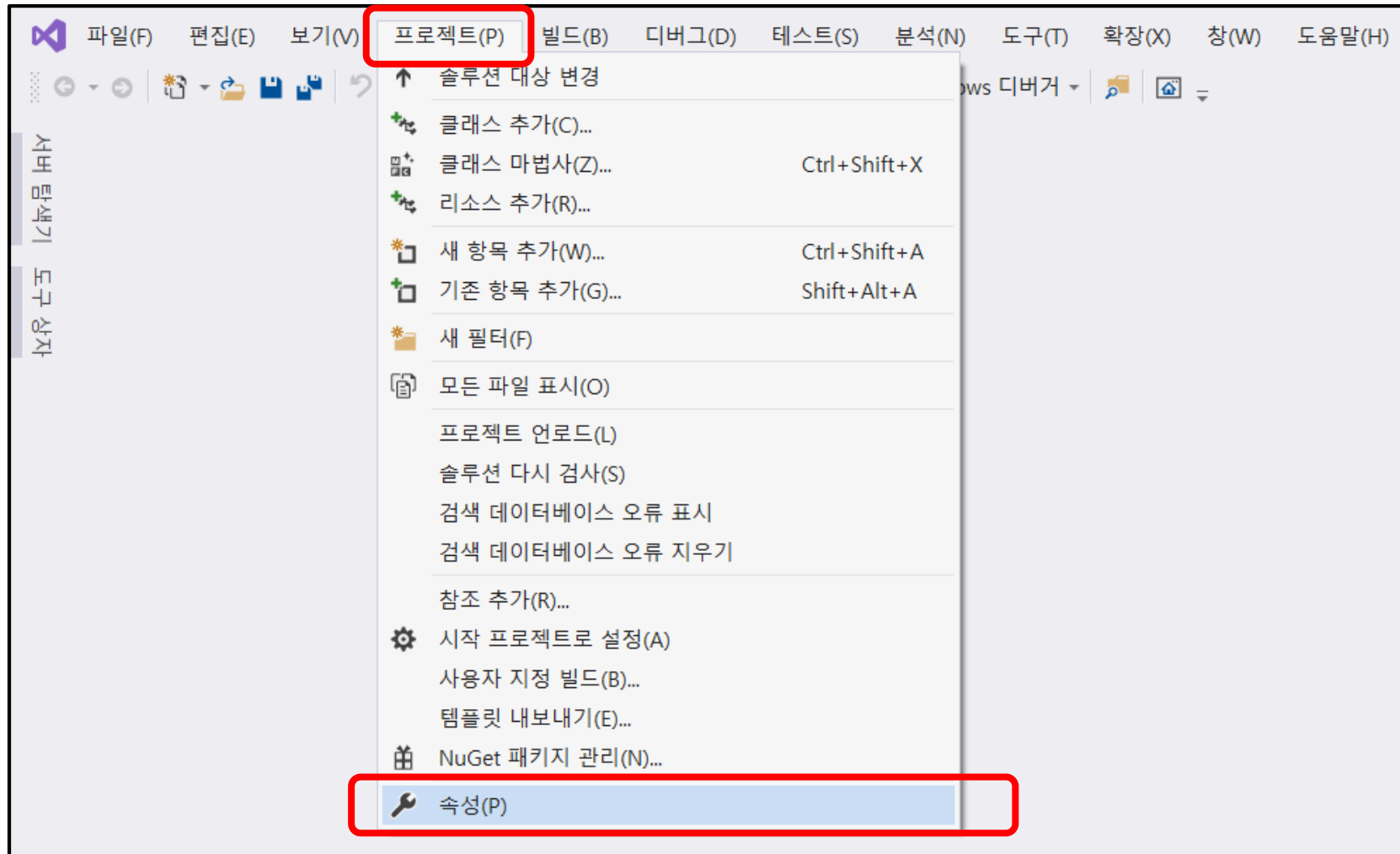


Visual Studio 설정하기

- Configuration 바꾸어 x64 플랫폼으로 설정하기
→ 빌드 > 구성관리자 메뉴에서 변경할 수도 있다.



Visual Studio 프로젝트 속성 설정



VS Project 속성 설정: 헤더파일 경로 설정

- 프로젝트 > 속성 > VC++ 디렉터리 > 포함 디렉터리에 다음 줄 추가

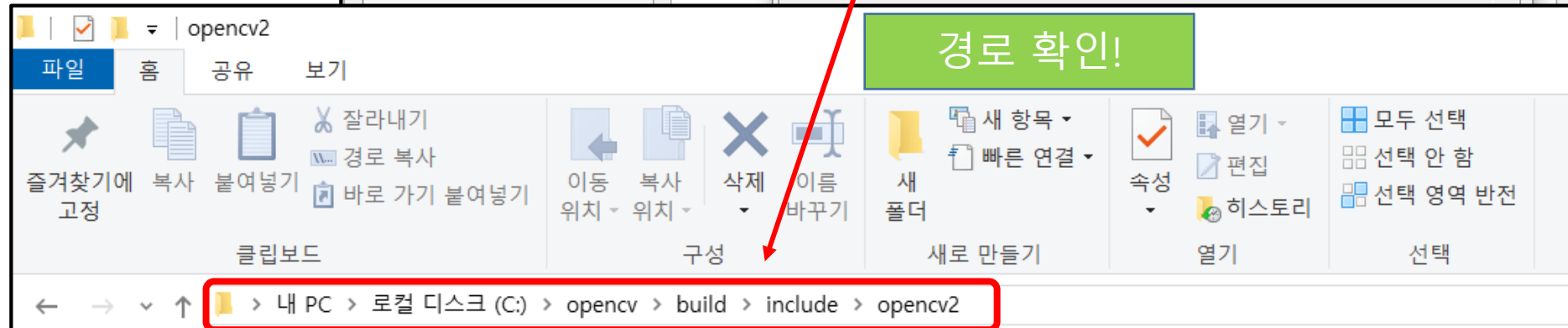
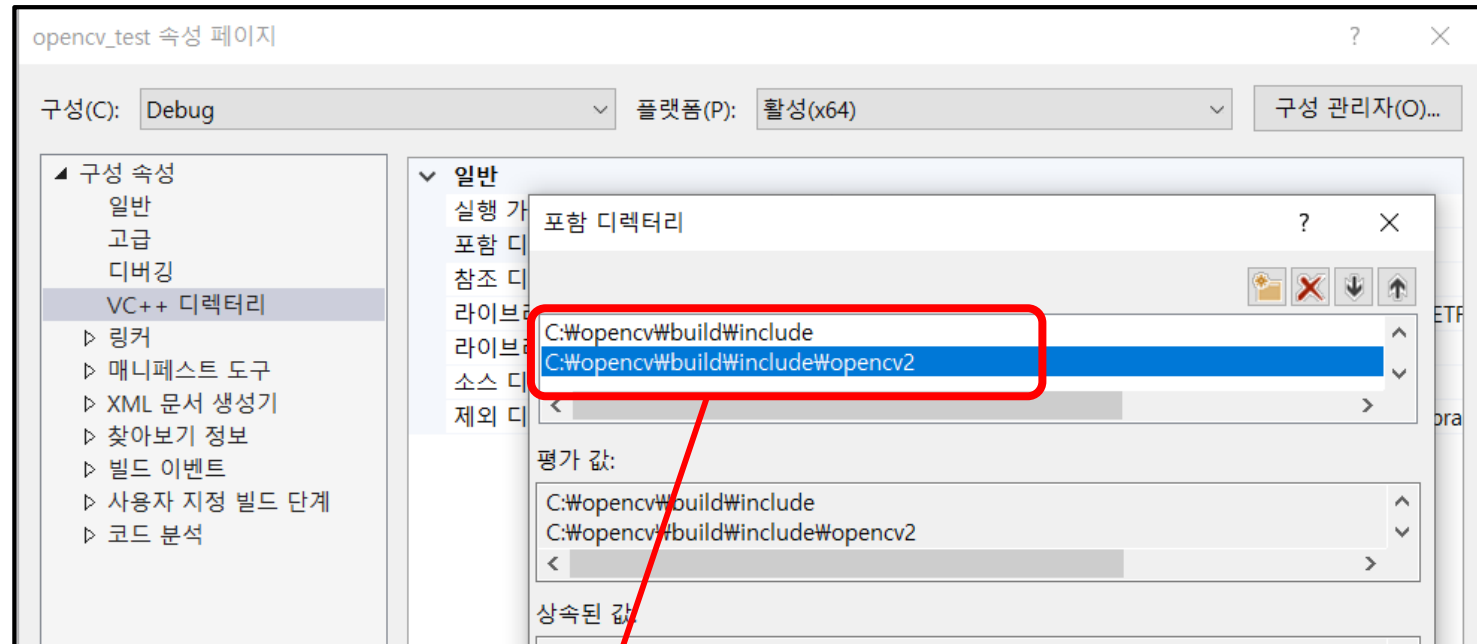
C:\Wopencv\build\include

C:\Wopencv\build\include\wopencv2

(두 항목은 세미콜론(;))으로 구별합니다)

- 추가한 뒤에 "적용"

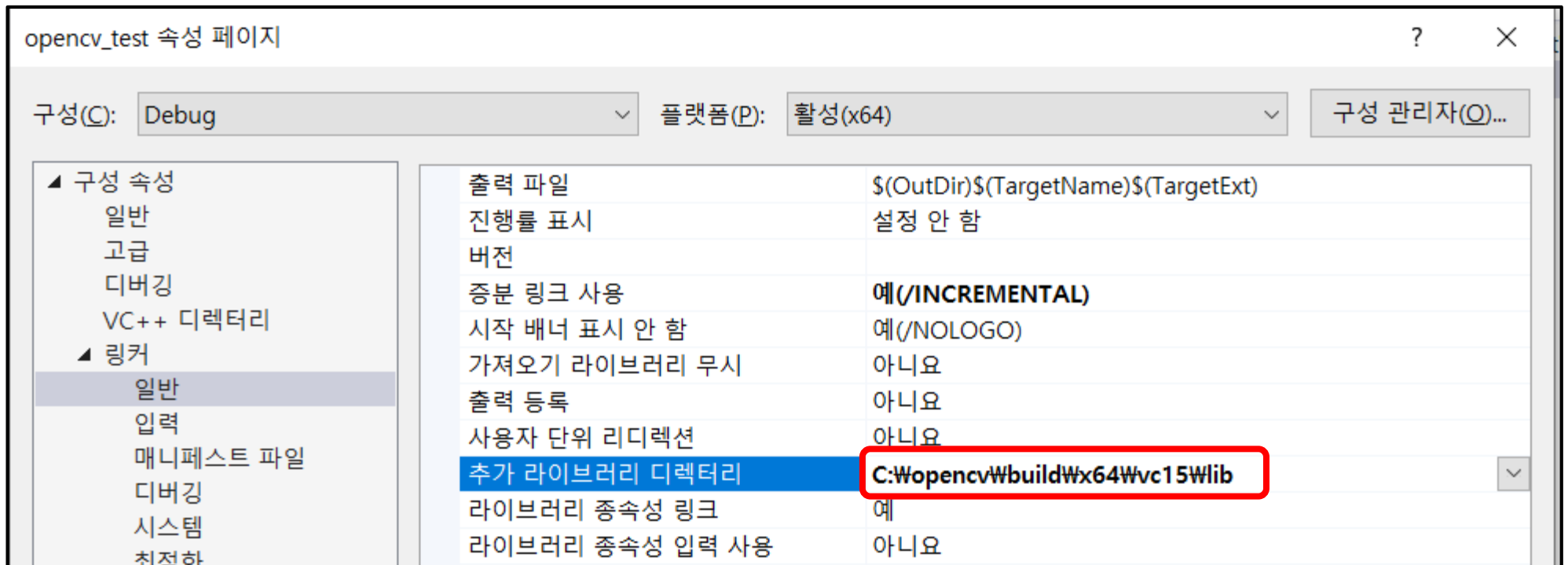
Opencv 설치된 경로
(사용자마다 다를 수 있으니 확인 필요)



VS Project 속성 설정: 추가 라이브러리 설정

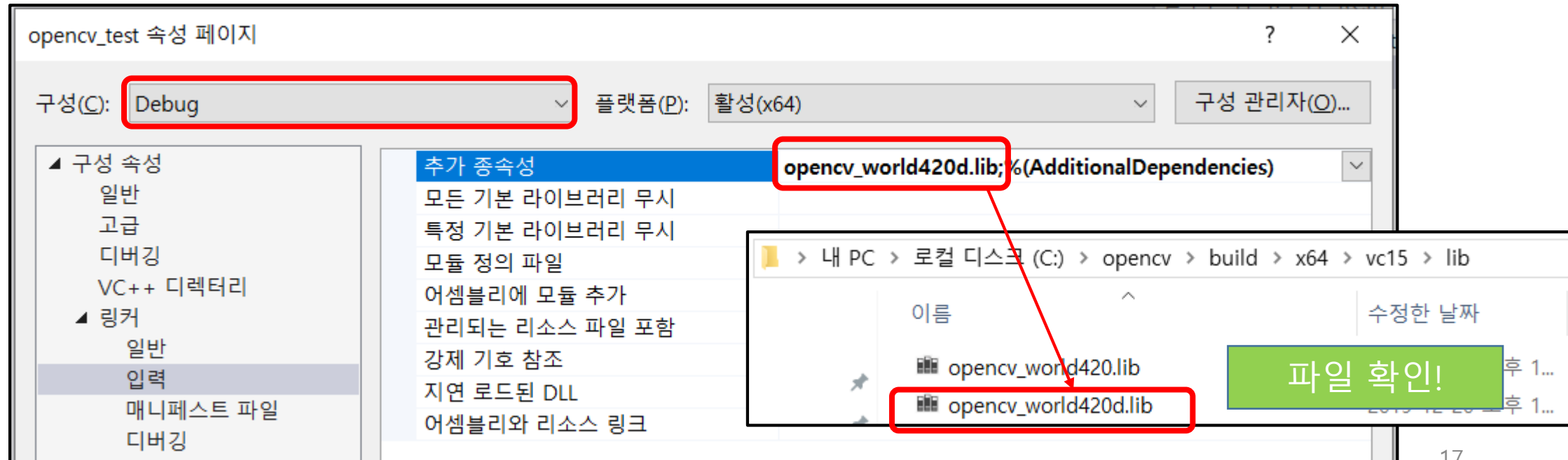
- 링커 > 일반 > 추가 라이브러리 디렉터리에 다음 줄 추가, 적용
C:\wopencv\build\x64\vc15\lib

Opencv 설치된
경로
(사용자마다 다를 수
있으니 확인 필요)



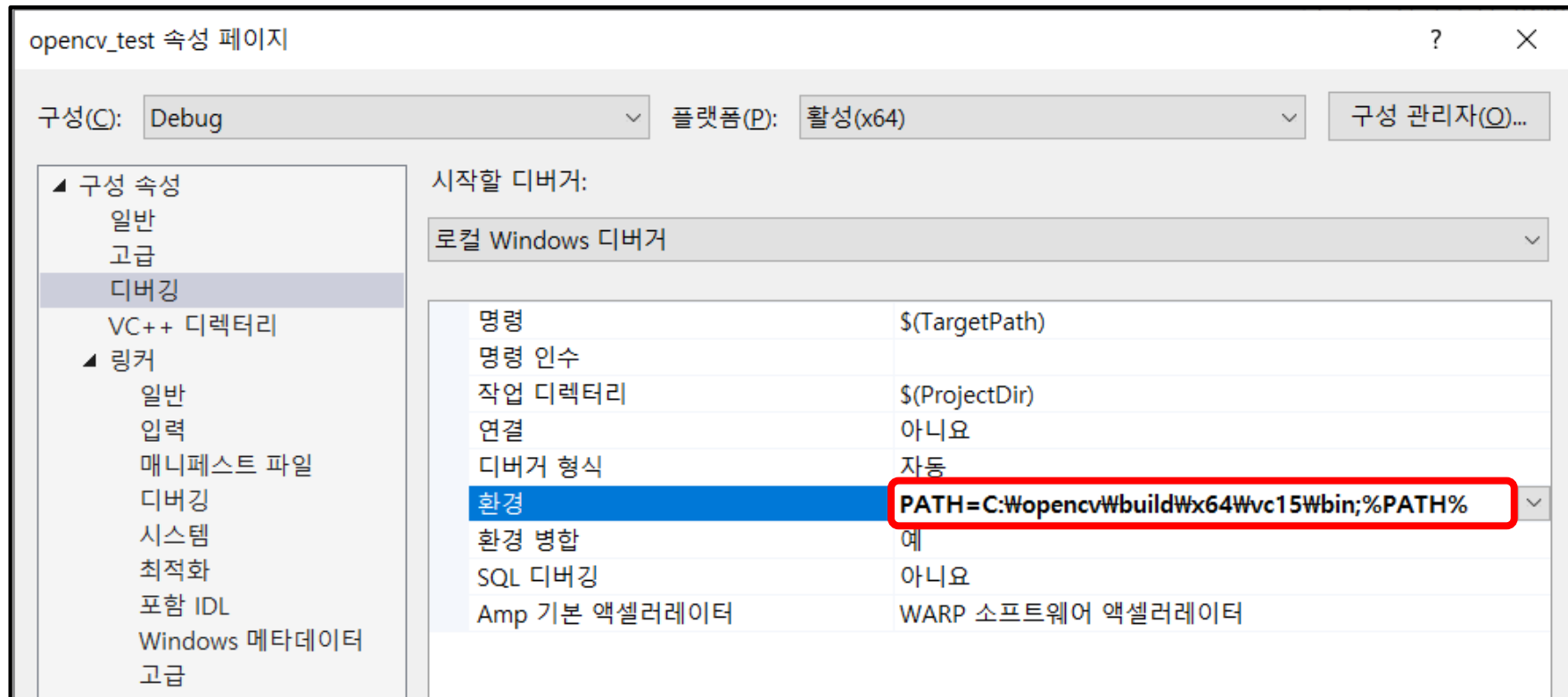
VS Project 속성 설정: 추가종속성 설정

- 링커 > 입력 > 추가종속성에 다음 줄을 추가하고 "적용"
opencv_world420d.lib (Debug용)
opencv_world420.lib (Release용)
주의) 위 두개의 라이브러리 파일을 같은 구성에 모두 추가하면 안됨!



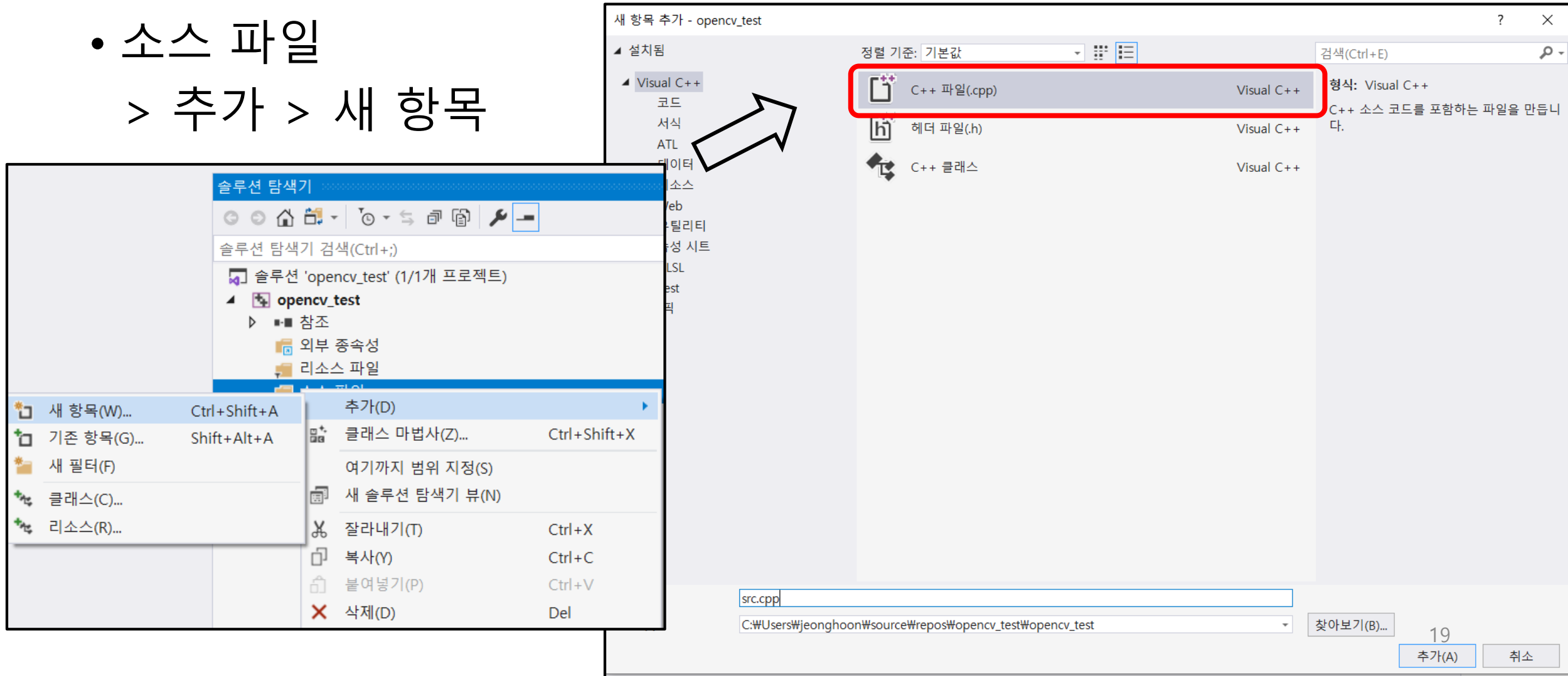
VS Project 속성 설정: 디버깅 설정

- 구성 속성 > 디버깅 > 환경에 다음 줄을 추가하고 “적용”
PATH=C:\wopencv\build\x64\vc15\bin;%PATH%



예제 코드 실행 : New file 생성

- 소스 파일
 - > 추가 > 새 항목



예제 코드 실행

클릭



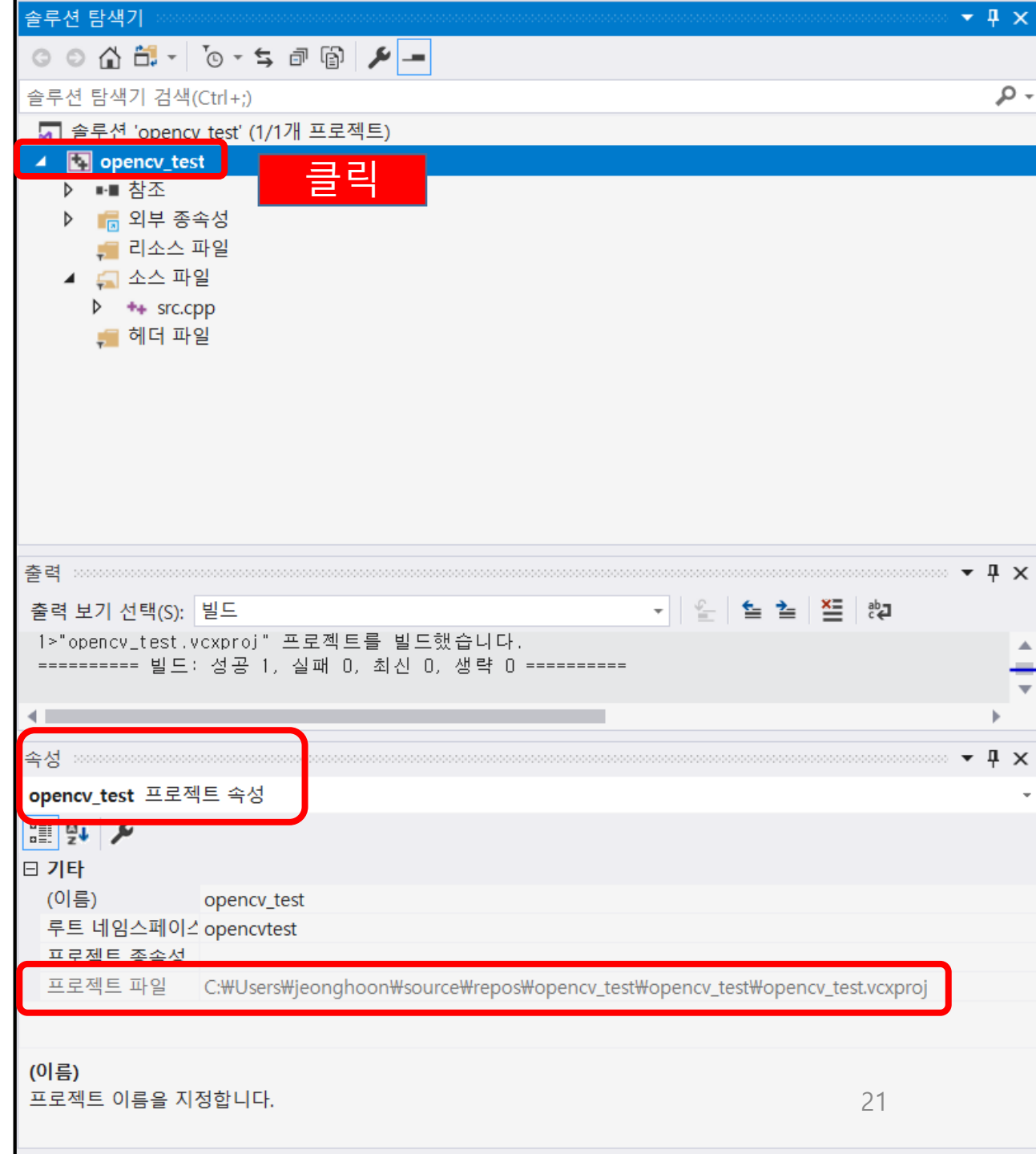
```
src.cpp* x
opencv_test (전역 범위)
1 #include <iostream>
2 #include <opencv2/core/core.hpp>
3 #include <opencv2/highgui/highgui.hpp>
4
5 using namespace cv;
6 using namespace std;
7
8 int main(int argc, const char* argv[]) {
9     Mat pic = imread("/Users/jeonghoon/Desktop/lena.png");
10    if (pic.empty()) {
11        cerr << "read fail!" << endl;
12        exit(-1);
13    }
14    imshow("Pic", pic);
15    waitKey(0);
16
17    return 0;
18 }
```

데스크탑에 있는 'lena.png'라는 예시 이미지 파일을 출력하는 코드입니다.
예시 코드 작성해 보실 때, 이 부분은 자유롭게,
가지고 있는 이미지 파일을 경로와 함께 넣어주시면 됩니다.
(프로젝트가 들어있는 폴더에 이미지 파일이 있는 경우, 이름만 넣으시면 됩니다.)

프로젝트 경로 확인

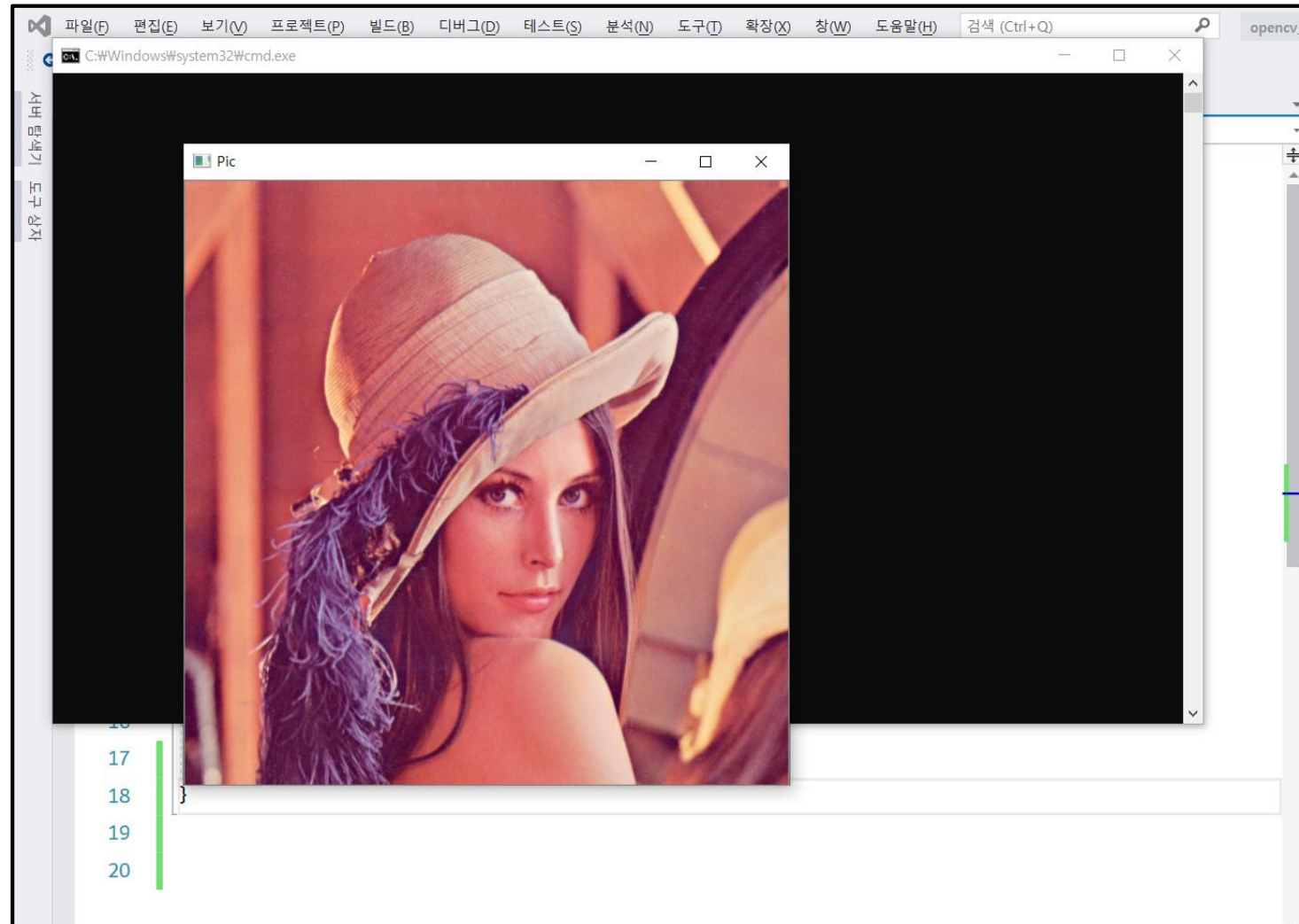
- 보기 > 속성 창
- 프로젝트를 클릭
- "프로젝트 파일"

(이 경로의 폴더에 읽고자 하는 파일을
넣으면 파일 이름만 넣어줘도
파일을 찾습니다.)



예제 코드 실행 결과 확인

아래 화면과 같이, 이미지 파일이 정상적으로 출력되면 문제없이 설치가 완료된 것입니다.



* OpenCV가 Visual Studio에서 실행이 안 될 때 check list

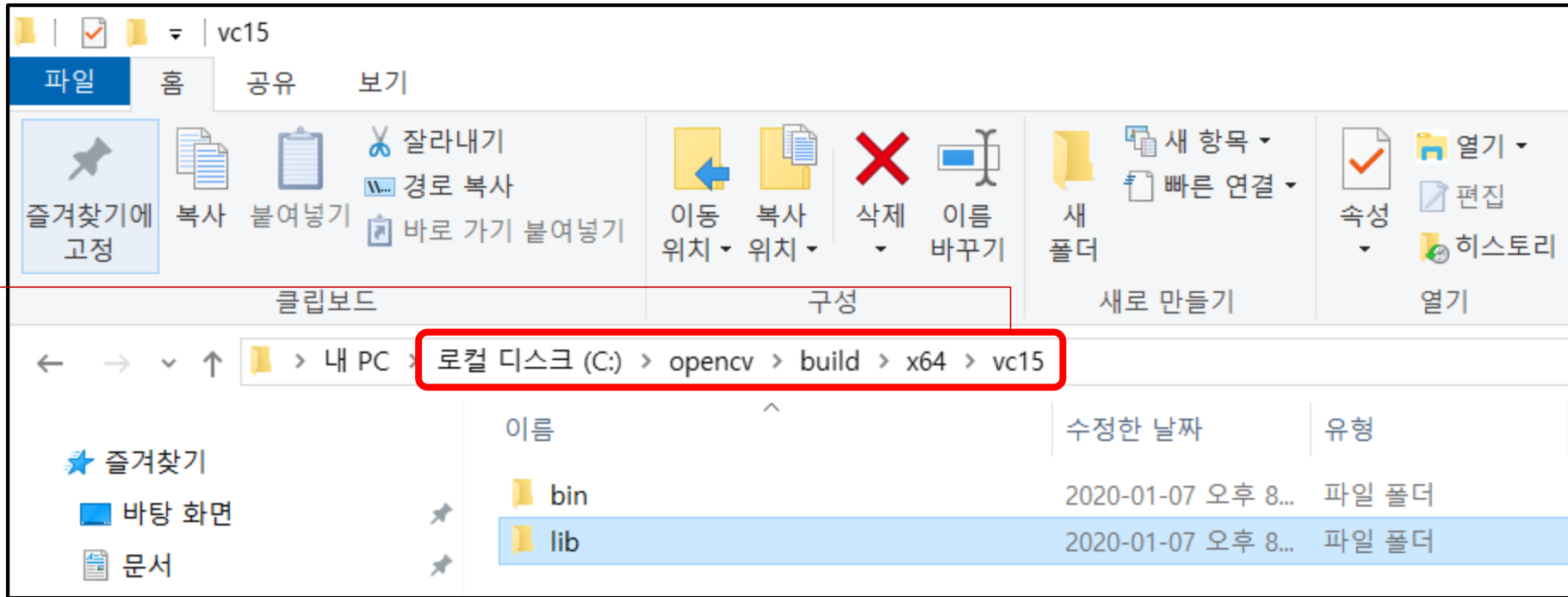
- Debug mode property에서 지정된 OpenCV의 lib 파일명에 d 글자가 포함되어 있는지 확인할 것(예: opencv**d.lib)
- Project의 folder 명 혹은 경로에 한글이 포함되지 않도록 할 것
- Debug mode를 Release mode로 바꾸어 볼 것
 - 주의) Release mode에서 Property를 설정해야 함

부록 1-2: Windows에서 OpenCV 사용하기

(opencv 설치위치를 windows 환경변수로 설정)

OpenCV 사용 환경 설정

: OpenCV 다운로드한 위치 확인



(opencv 설치 위치가 C:\wopencv이고, 64-bit 윈도우에서 Visual Studio 2017 또는 Visual Studio 2019를 쓴다고 가정)

→ *주의 **vc15**가 **Visual Studio 2017 또는 Visual Studio 2019**에 해당하는 컴파일러 버전이다.
vc14는 **Visual Studio 2015** 컴파일러 버전이다.

OpenCV 사용 환경 설정

: 환경변수 (OPENCV_DIR) 설정

- 관리자 권한으로 cmd 창을 열어서 다음 명령어 실행
> setx -m OPENCV_DIR C:\opencv\build\x64\vc15



```
C:\Windows\system32>setx -m OPENCV_DIR C:\opencv\build\x64\vc15
성공: 지정한 값을 저장했습니다.
```

(opencv 설치 위치가 C:\opencv이고, 64-bit 윈도우에서 Visual Studio 2017 또는 Visual Studio 2019를 쓴다고 가정)

*주의 **vc15**가 **Visual Studio 2017** 또는 **Visual Studio 2019**에 해당하는 컴파일러 버전이다.
vc14는 **Visual Studio 2015** 컴파일러 버전이다.

OpenCV 사용 환경 설정

: 환경변수 (OPENCV_DIR) 설정

- 관리자 권한으로 cmd 창을 다시 열어 환경 변수 설정이 잘 되었나 확인
>set

```
C:\Windows\system32>set
ALLUSERSPROFILE=C:\ProgramData
APPDATA=C:\Users\jeonghoon\AppData\Roaming
CommonProgramFiles=C:\Program Files\Common Files
CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
CommonProgramW6432=C:\Program Files\Common Files
COMPUTERNAME=DESKTOP-A10GNL3
ComSpec=C:\Windows\system32\cmd.exe
DriverData=C:\Windows\System32\Drivers\DriverData
HOMEDRIVE=C:
HOMEPATH=\Users\jeonghoon
LOCALAPPDATA=C:\Users\jeonghoon\AppData\Local
LOGONSERVER=\\DESKTOP-A10GNL3
NUMBER_OF_PROCESSORS=8
OneDrive=C:\Users\jeonghoon\OneDrive
OPENCV_DIR=C:\opencv\build\x64\vc15
OS=Windows_NT
```

Visual Studio (VS) 설정하기

- Configuration 바꾸어 x64 플랫폼으로 설정하기
- openCV 라이브러리 다운로드 받기
- openCV 라이브러리 및 헤더파일 경로 설정하기

→ 다음 슬라이드를 보면서 설정하세요!

Visual Studio : New project 생성



Visual Studio : New project 생성

새 프로젝트 구성

Windows 데스크톱 마법사 C++ Windows 데스크톱 콘솔 라이브러리

프로젝트 이름(N)
opencv_test

위치(L)
C:\Users\Wjeonghoon\source\repos

솔루션 이름(M)
opencv_test

☐ 솔루션 및 프로젝트를 같은 디렉터리

Windows 데스크톱 프로젝트

애플리케이션 종류(T)
콘솔 애플리케이션(.exe)

추가 옵션:
☒ 빈 프로젝트(E)
☐ 미리 컴파일된 헤더(P)
☐ 기호 내보내기(X)
☐ MFC 헤더(M)

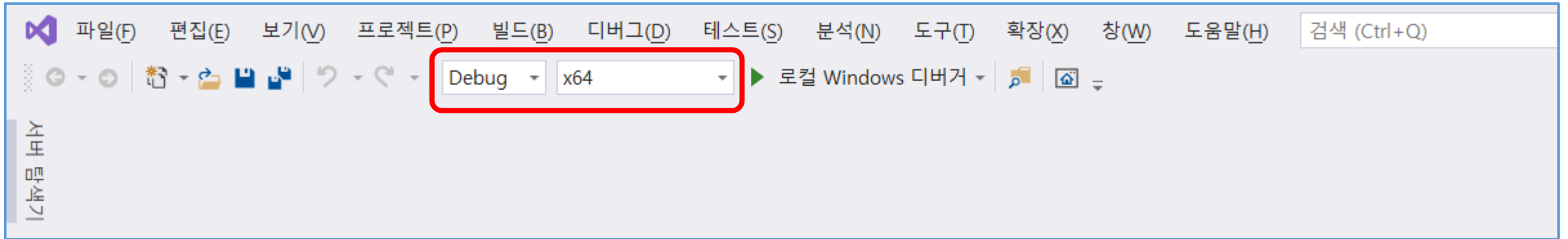
팁: 빈 프로젝트 템플릿을 사용하여 이 종류의 프로젝트를 만들 수도 있습니다.

확인 취소

뒤로(B) 만들기(C)

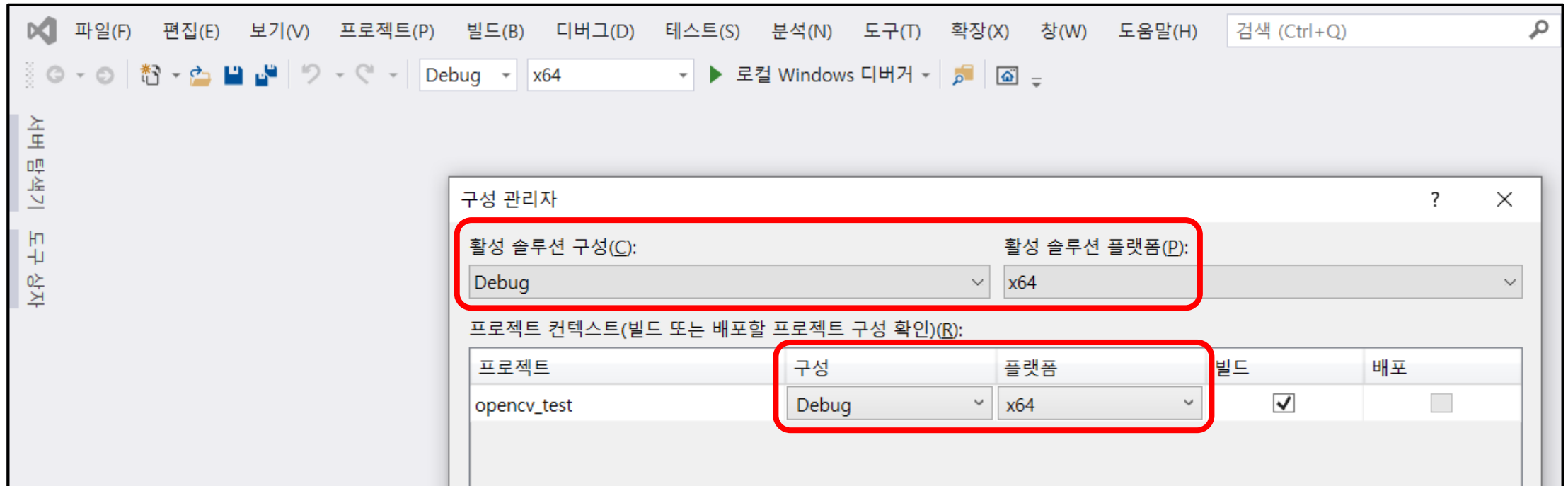
Visual Studio 설정하기

- Configuration 바꾸어 x64 플랫폼으로 설정하기

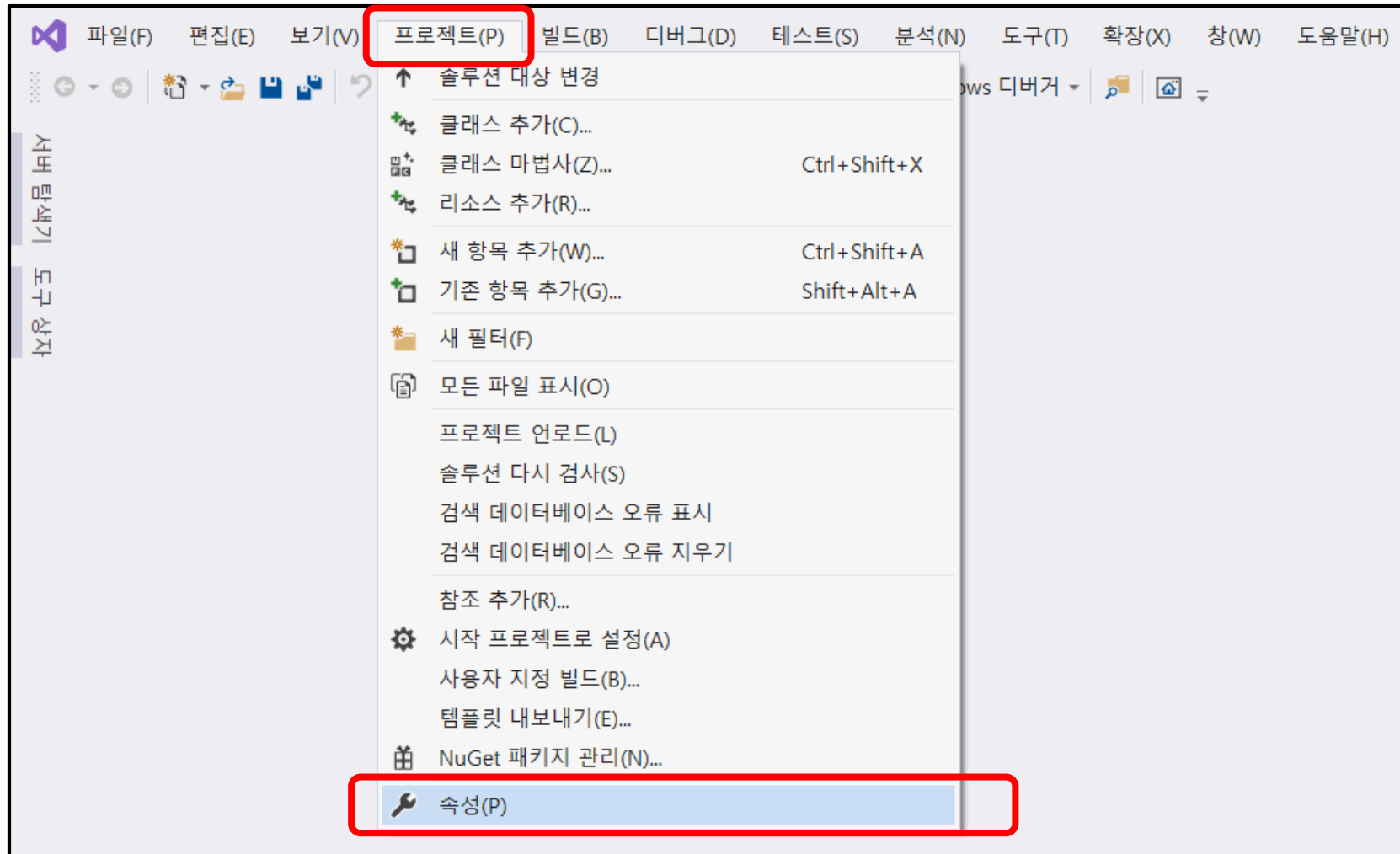


Visual Studio 설정하기

- Configuration 바꾸어 x64 플랫폼으로 설정하기
→ 빌드 > 구성관리자 메뉴에서 변경할 수도 있다.



Visual Studio 프로젝트 속성 설정



VS Project 속성 설정: 헤더파일 경로 설정

- 프로젝트 > 속성 > VC++ 디렉터리

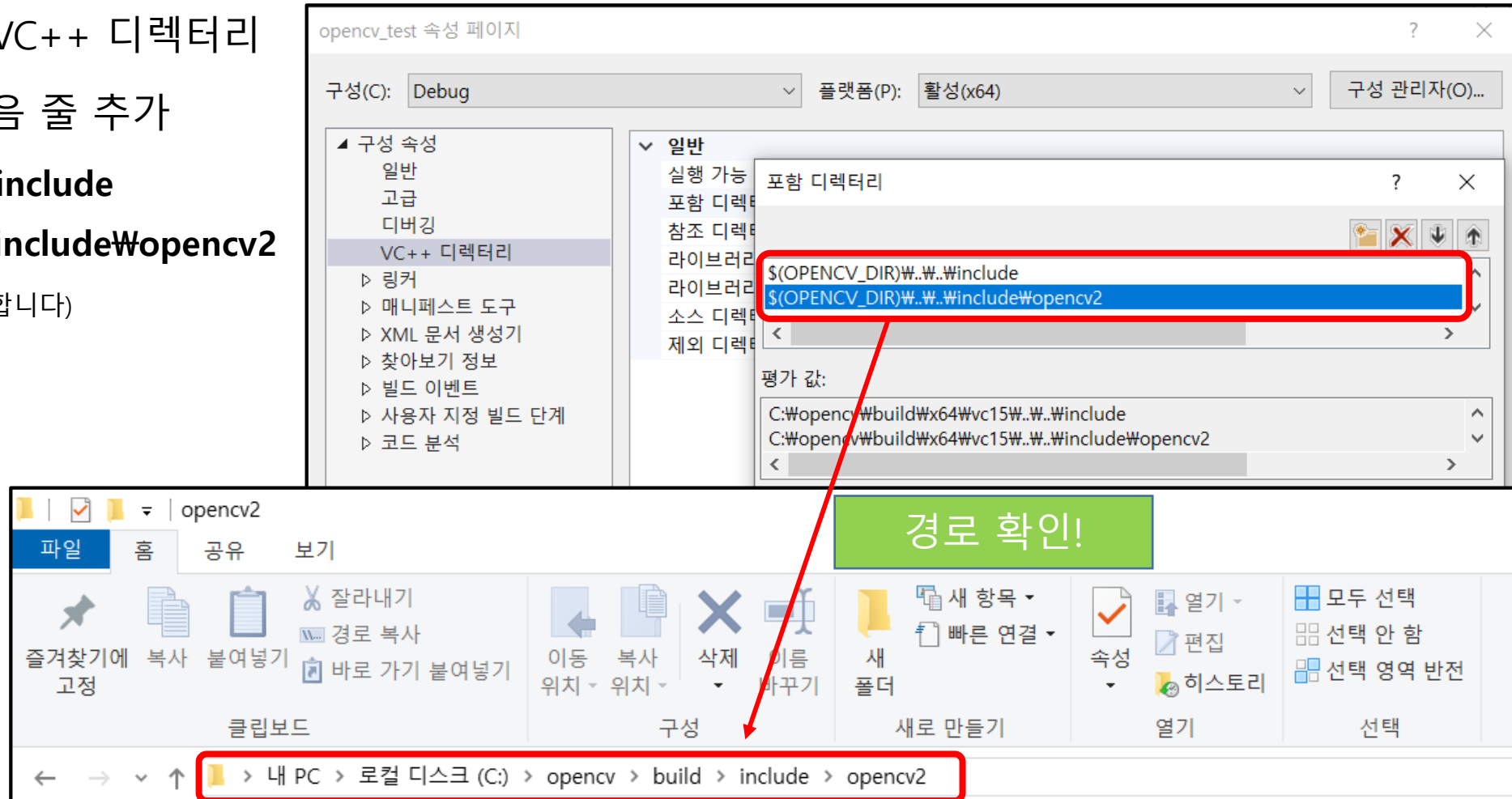
- > 포함 디렉터리에 다음 줄 추가

`$(OPENCV_DIR)..\..\Wininclude`

`$(OPENCV_DIR)..\..\Wininclude\opencv2`

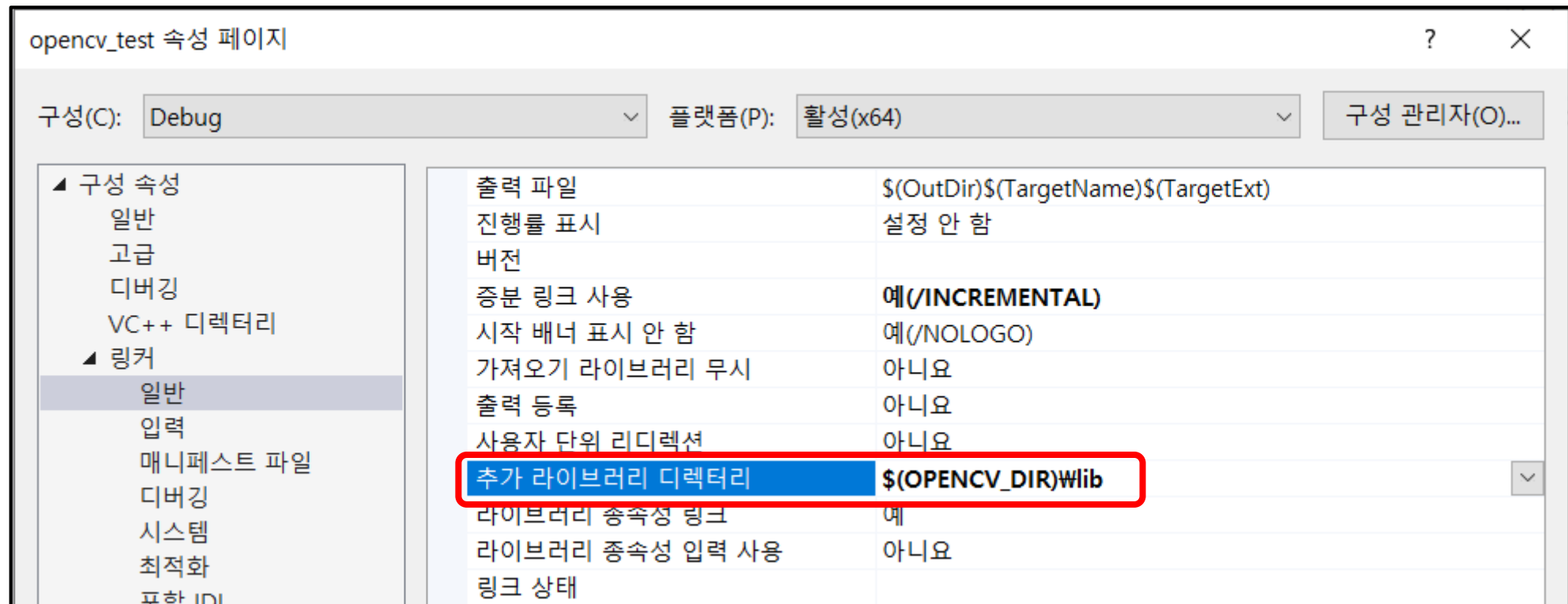
(두 항목은 세미콜론(;))으로 구별합니다)

- 추가한 뒤에 "적용"



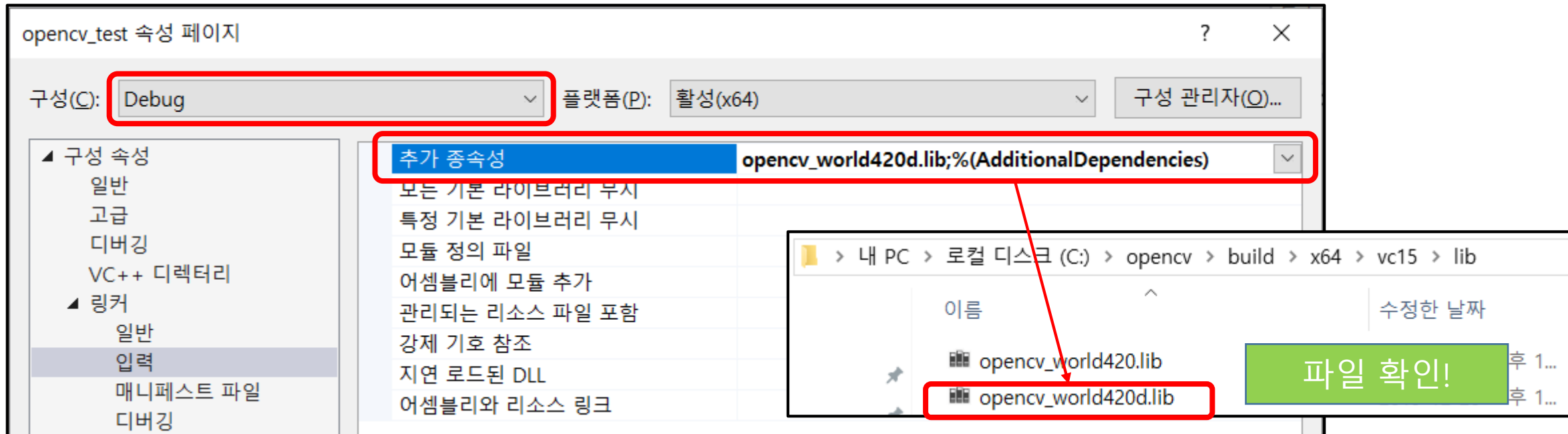
VS Project 속성 설정: 추가 라이브러리 설정

- 링커 > 일반 > 추가 라이브러리 디렉토리에 다음 줄 추가, 적용
\$(OPENCV_DIR)\lib



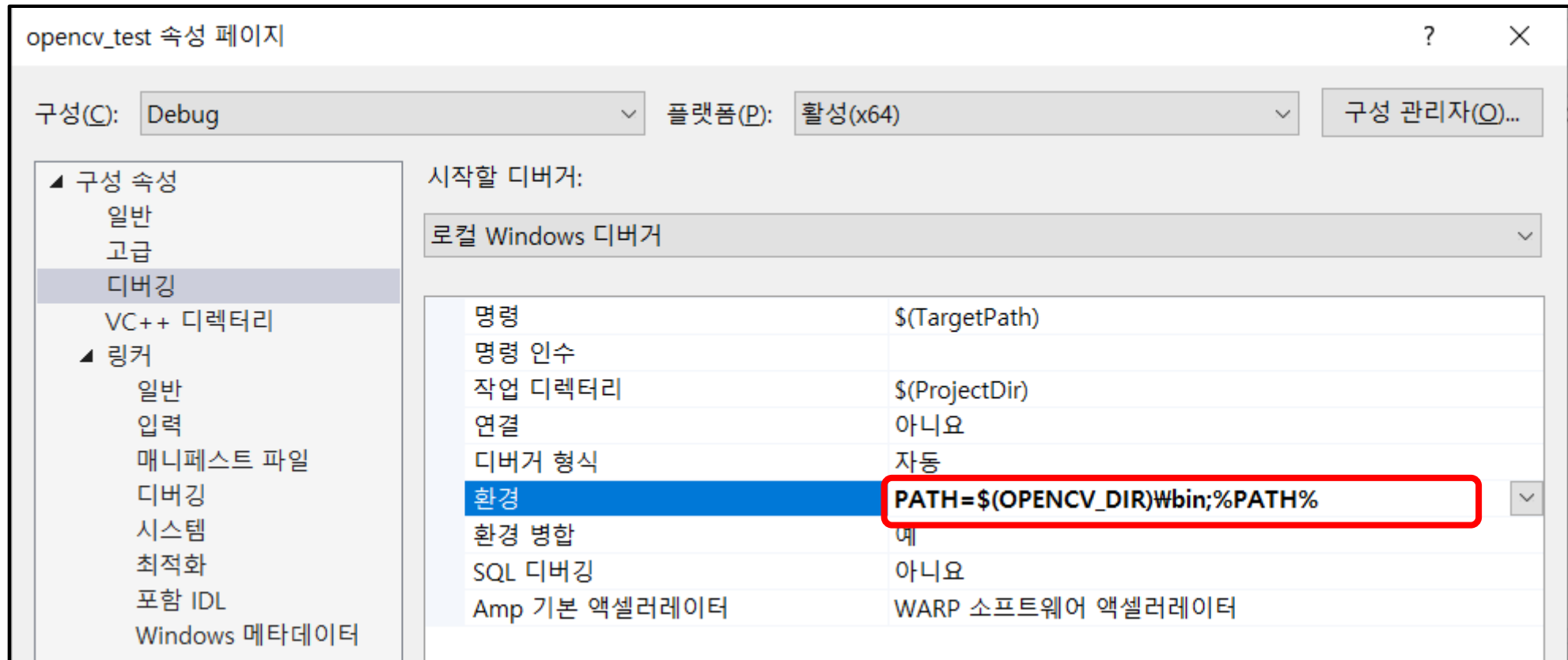
VS Project 속성 설정: 추가종속성 설정

- 링커 → 입력 → 추가종속성에 다음 줄을 추가하고 “적용”
opencv_world420d.lib (Debug용)
opencv_world420.lib (Release용)



VS Project 속성 설정: 디버깅 설정

- 구성 속성 > 디버깅 > 환경에 다음 줄을 추가하고 “적용”
PATH=\$(OPENCV_DIR)\bin;%PATH%



부록2: Mac OS X에서 OpenCV 사용하기

1. 홈브루(Homebrew)로 OpenCV 설치

- 홈브루(Homebrew)가 설치되어 있지 않다면, 먼저 **homebrew**를 설치해줍니다.
- 아래 명령어를 복사하여 **터미널**에 입력하면 **homebrew**설치가 시작됩니다.

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

- 설치가 완료되면 이어서 homebrew로 **opencv**를 설치합니다.

터미널에,

```
brew update
```

```
brew install opencv
```

```
brew upgrade opencv
```

명령어를 차례로 입력하여 **opencv**를 설치합니다.

2. Xcode 설치

<https://developer.apple.com/xcode/download> 사이트에서 로그인 한 뒤 Xcode 11.3을 설치해줍니다.

Beta Software Downloads



Xcode 11.3

Xcode 11.3 includes everything you need to create amazing apps for all Apple platforms. Includes the latest SDKs for macOS, iOS, watchOS, and tvOS.

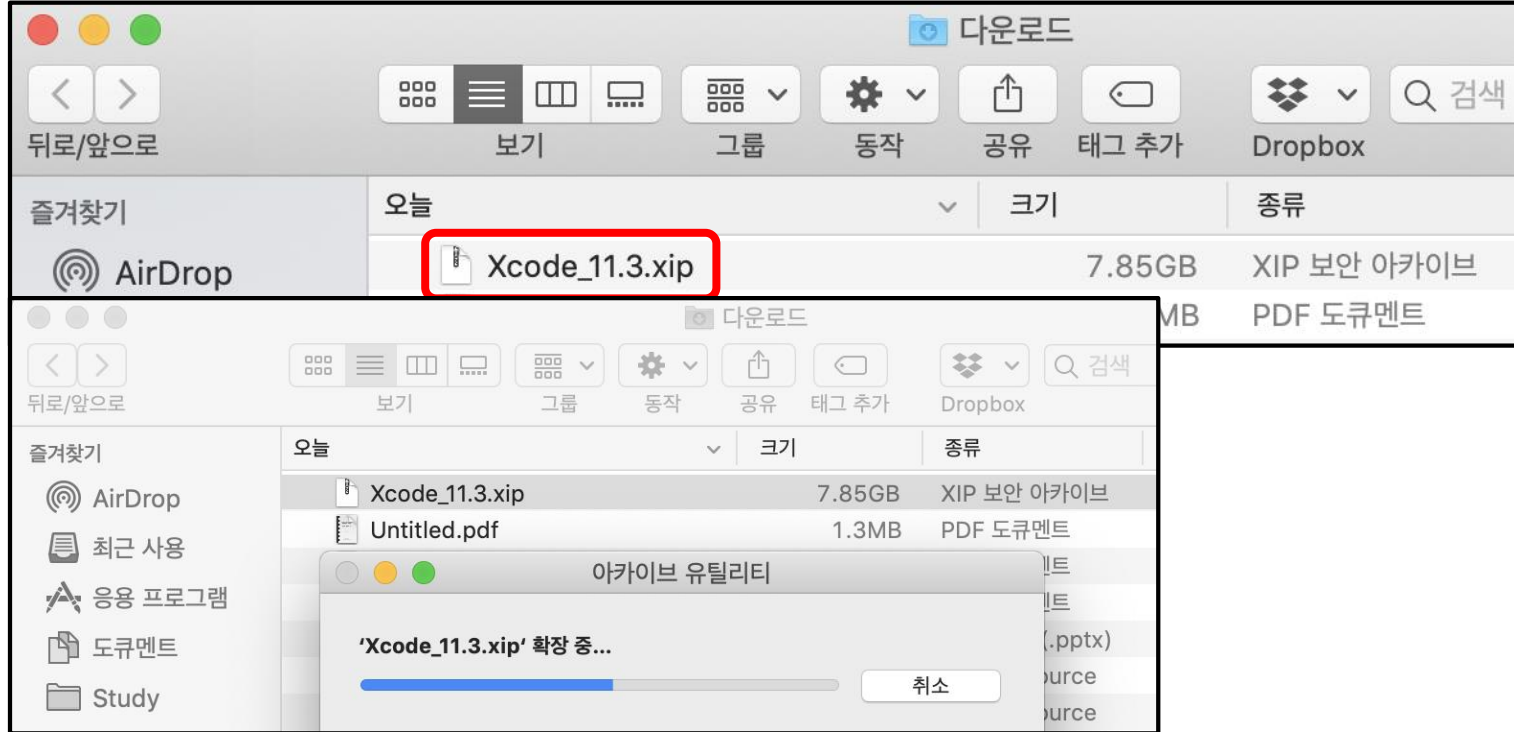
[Download](#)[Release Notes >](#)

Released
December 10, 2019

Build
11C29

2. Xcode 설치

다운로드 받은 Xcode_11.3.xip를 "더블클릭"하여 xip파일을 열어 줍니다.

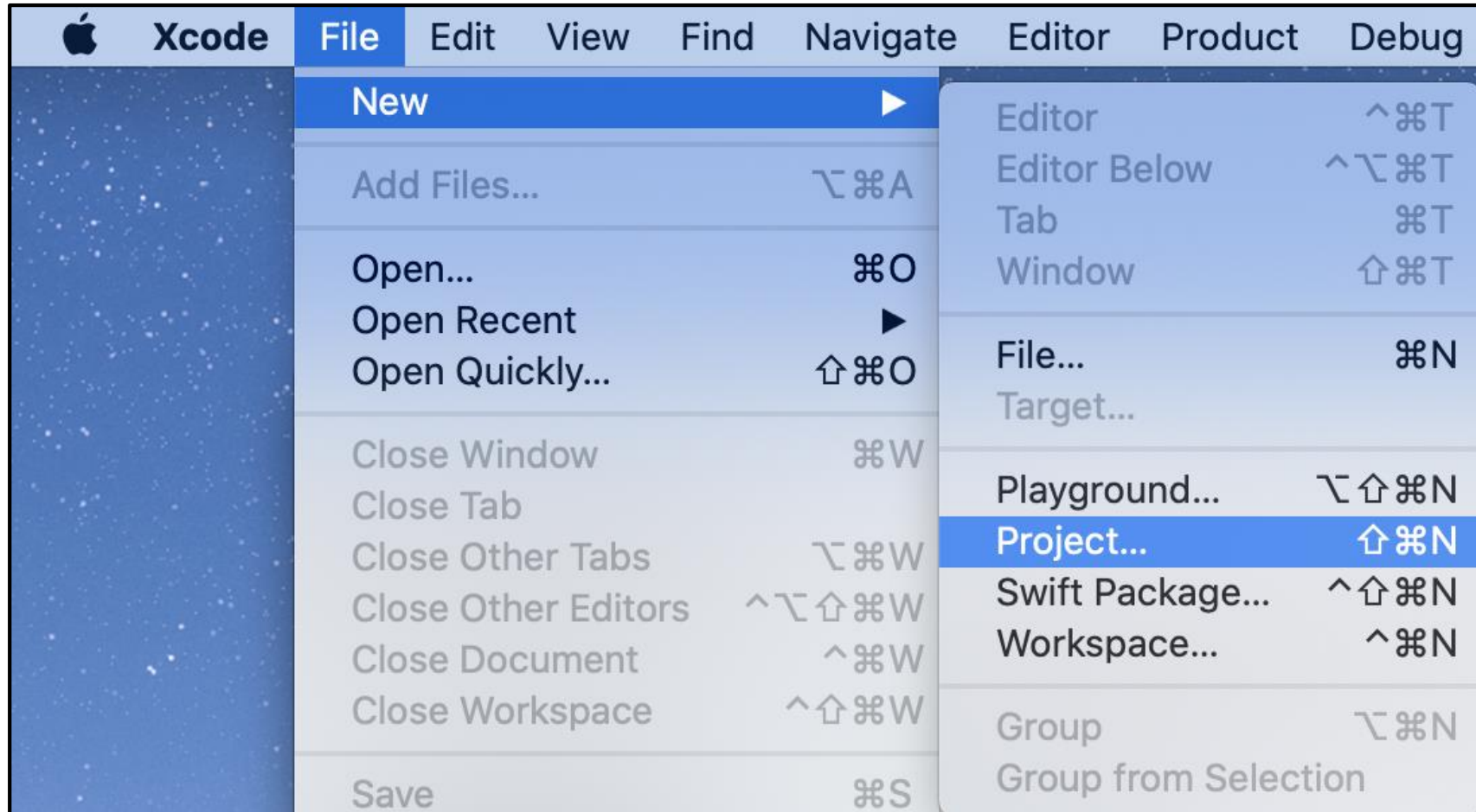


Xcode를 실행하기 위해 OS 버전
업데이트 등 시간이 소요될 수 있
으니 주의하세요!

위치		
원격 디스크 네트워크	2019	
	Zoom.pkg	12.1MB 설치 프...램 패키지
	Xcode	16.17GB 응용 프로그램

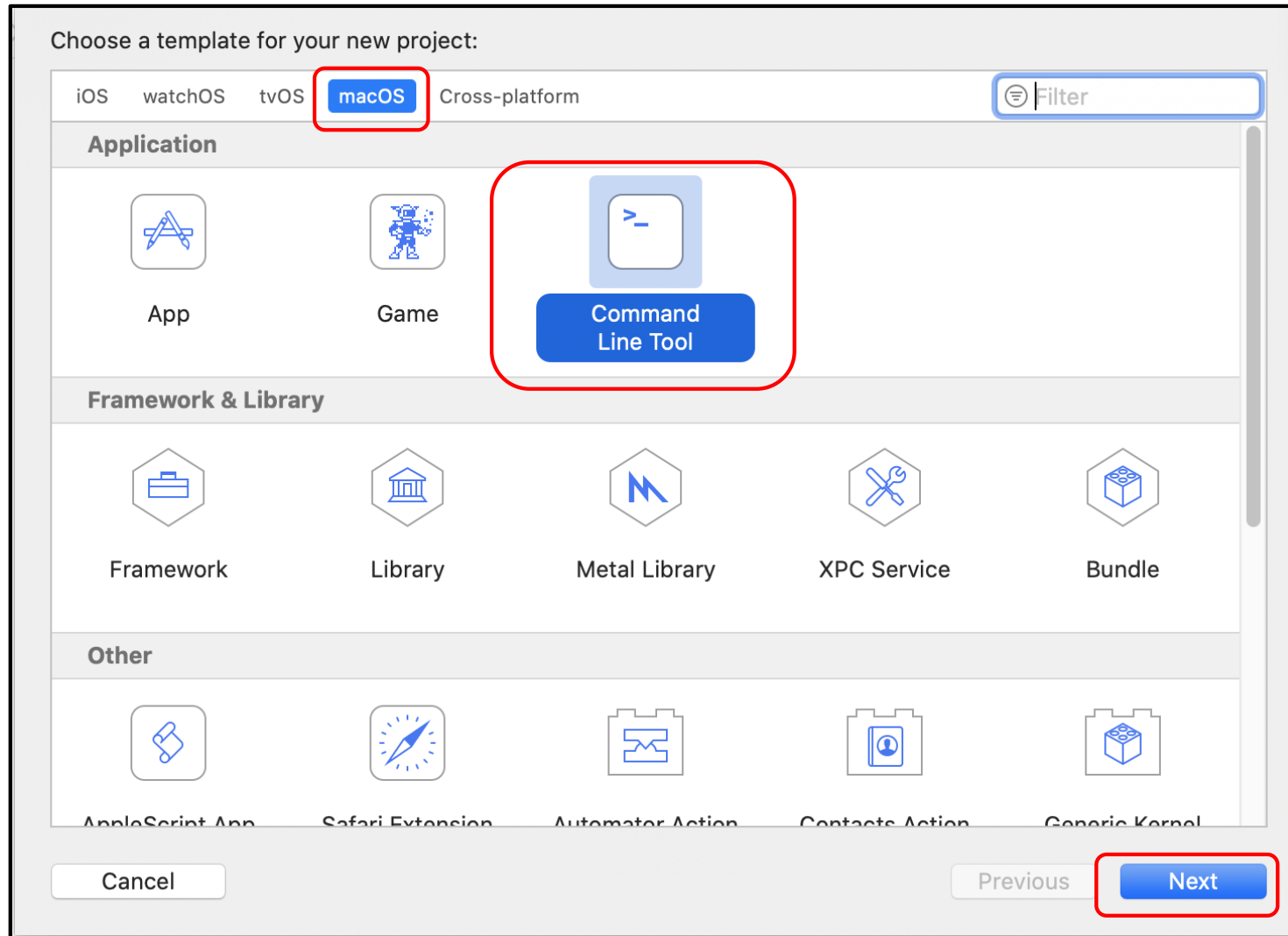
3. Xcode 에서 프로젝트 생성

설치가 완료되면 Xcode를 실행한 후, File > New > Project를 클릭합니다.



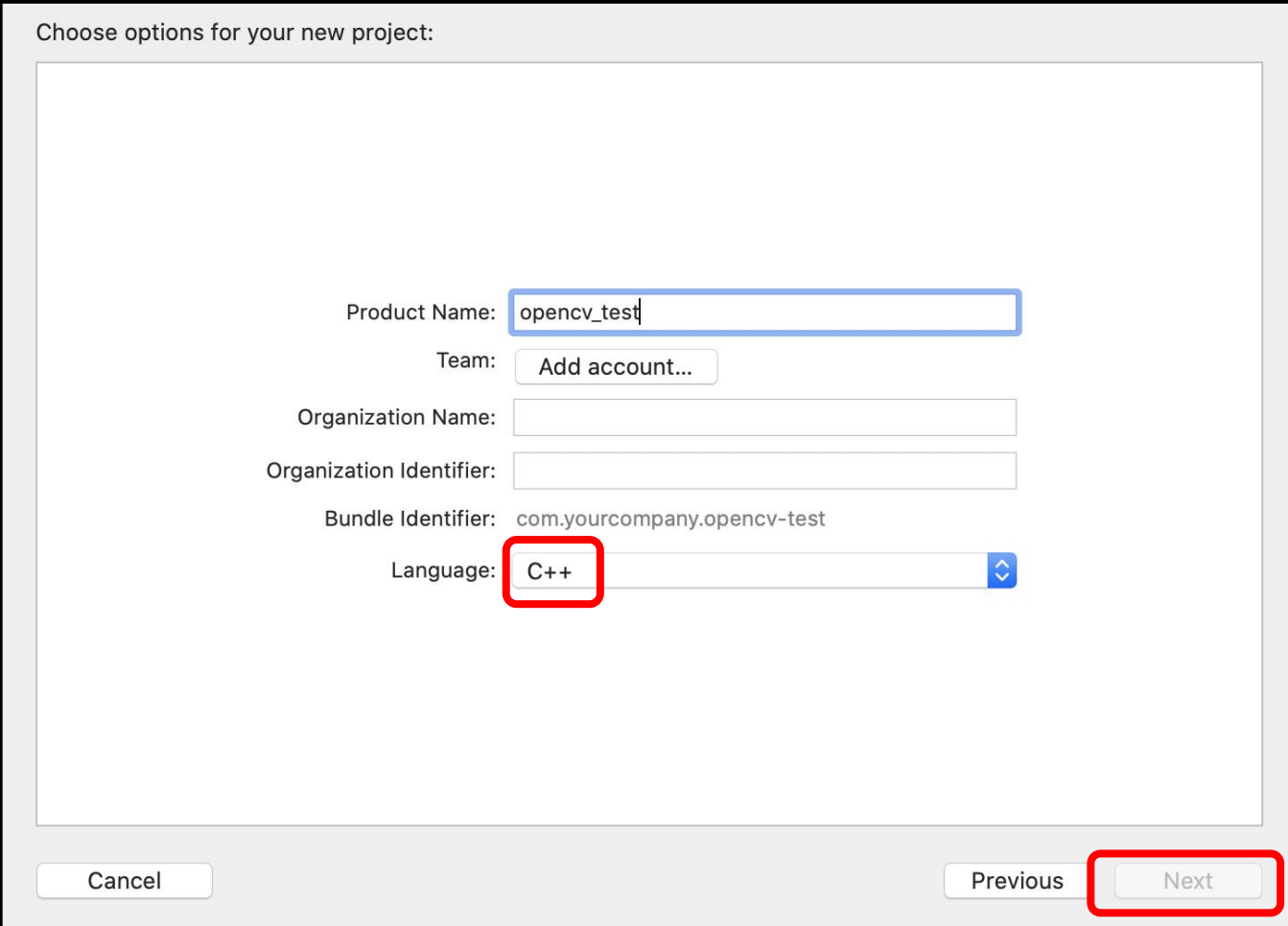
3. Xcode 에서 프로젝트 생성

'macOS → command Line Tool → Next' 차례대로 클릭



3. Xcode 에서 프로젝트 생성, 환경 설정

원하는 'Product Name' 입력, 'Language' → 'C++' 로 설정한 후 'Next' 클릭



The image shows the 'Choose options for your new project' dialog box in Xcode. The dialog has a title bar and a main content area. The content area contains several input fields and a dropdown menu. The 'Product Name' field is filled with 'opencv_test'. The 'Team' field has a button 'Add account...'. The 'Organization Name' and 'Organization Identifier' fields are empty. The 'Bundle Identifier' field is filled with 'com.yourcompany.opencv-test'. The 'Language' dropdown menu is set to 'C++'. At the bottom of the dialog, there are three buttons: 'Cancel', 'Previous', and 'Next'. The 'Next' button is highlighted with a red rectangle.

Choose options for your new project:

Product Name: opencv_test

Team: Add account...

Organization Name:

Organization Identifier:

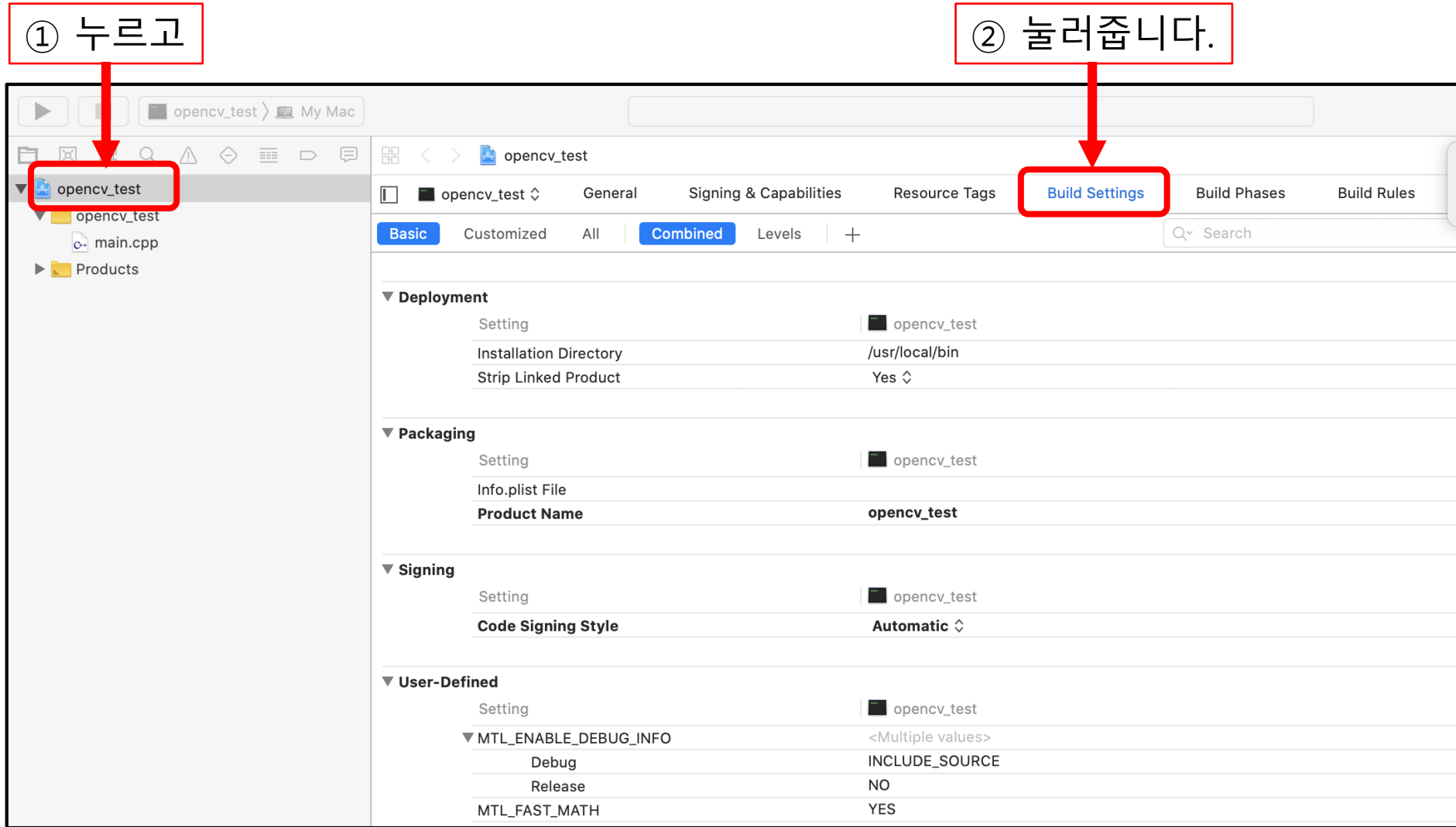
Bundle Identifier: com.yourcompany.opencv-test

Language: C++

Cancel Previous Next

4. Xcode 에서 프로젝트 환경 설정

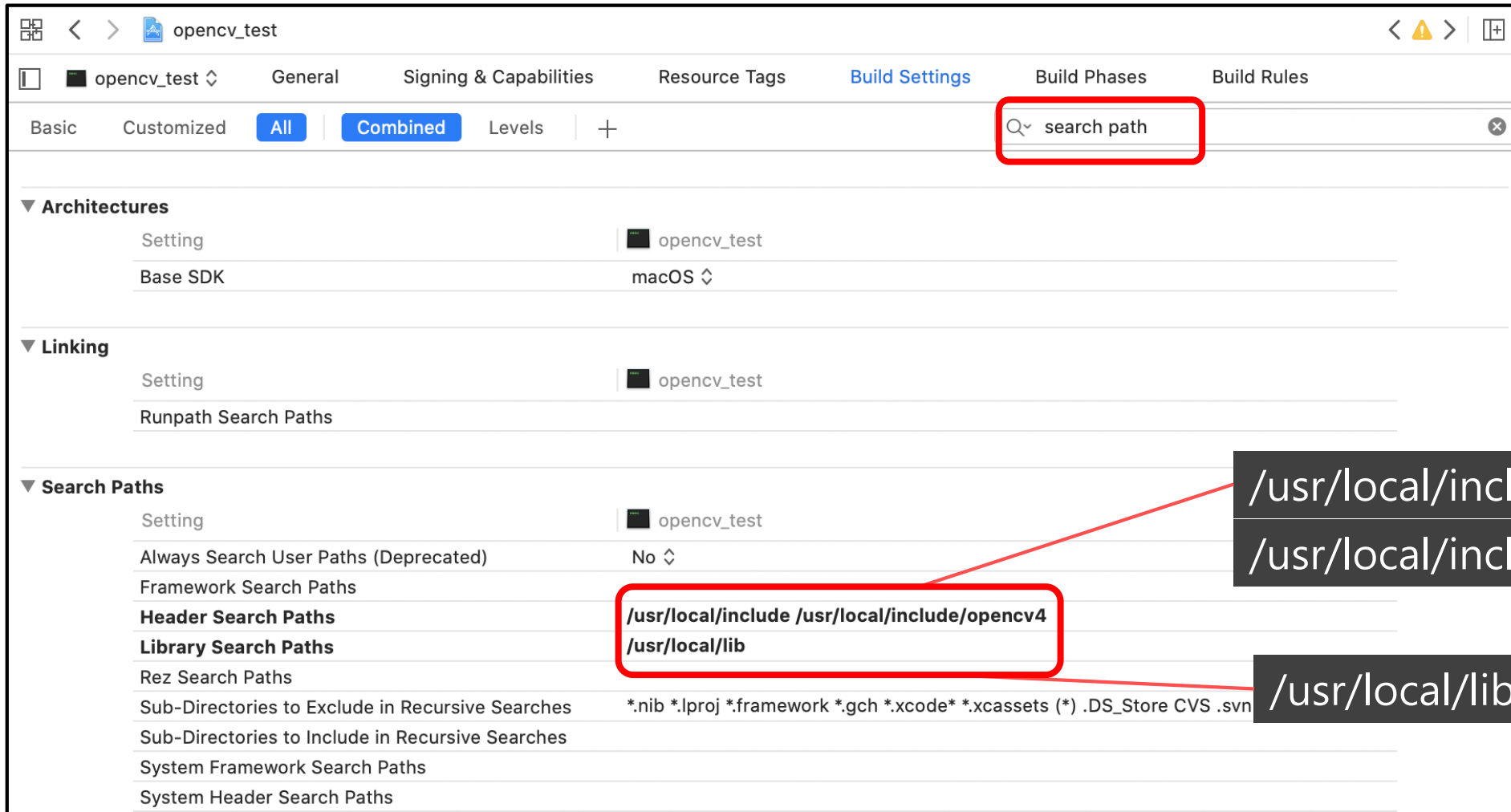
생성된 프로젝트의 환경 설정을 위해 **Build Settings** 탭을 눌러줍니다.



4. Xcode 에서 프로젝트 환경 설정

검색창에 'search path' 검색하여

'Search Paths → Header Search Paths, Library Search Paths' 에 다음과 같이 입력



/usr/local/include

/usr/local/include/opencv4

/usr/local/lib

4. Xcode 에서 프로젝트 환경 설정

- Linker Flags 설정값을 알아내기 위해 먼저 설치된 opencv 버전을 알아봅니다.
터미널에서 아래 명령어 입력한 결과가 **opencv 버전**입니다.

```
ls /usr/local/Cellar/opencv
```

Ex)

```
$ ls /usr/local/Cellar/opencv  
4.2.0_1
```

- 다음으로 아래 명령어를 터미널에 입력한 결과를 복사해줍니다.

```
pkg-config --cflags --libs /usr/local/Cellar/opencv/(설치된 opencv 버전)/lib/pkgconfig/opencv4.pc
```

4.2.0_1

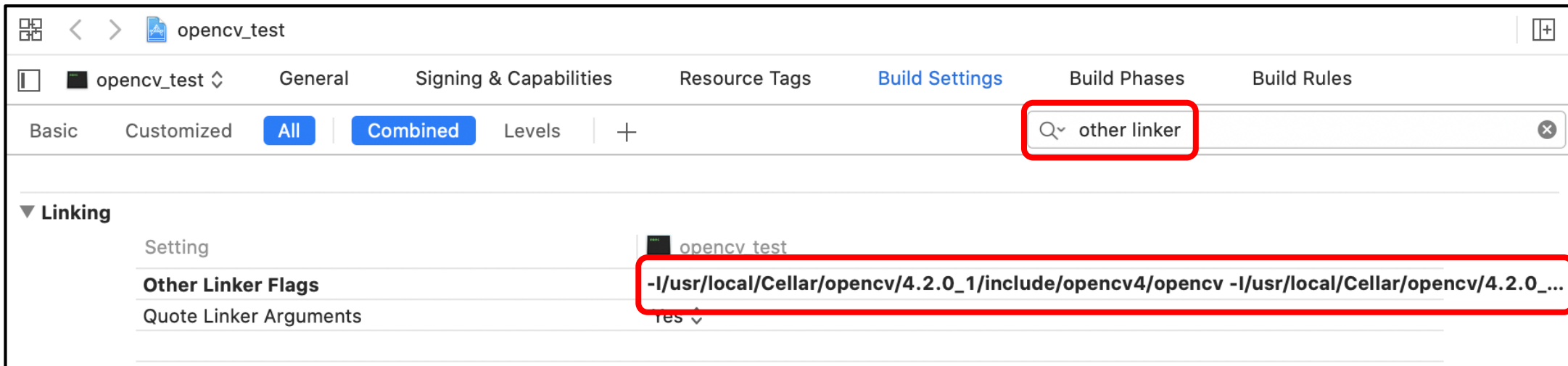
Ex)

```
$ pkg-config --cflags --libs /usr/local/Cellar/opencv/4.2.0_1/lib/pkgconfig/opencv4.pc  
-I/usr/local/Cellar/opencv/4.2.0_1/include/opencv4/opencv -I/usr/local/Cellar/opencv/4.2.0_1/include/opencv4 -L/usr/local/Cellar/opencv/4.2.0_1/lib -lopencv_gapi -lopencv_stitching -lopencv_aruco -lopencv_bgsegm -lopencv_bioinspired -lopencv_ccalib -lopencv_dnn_objdetect -lopencv_dnn_superres -lopencv_dpm -lopencv_highgui -lopencv_face -lopencv_freetype -lopencv_fuzzy -lopencv_hfs -lopencv_img_hash -lopencv_line_descriptor -lopencv_quality -lopencv_reg -lopencv_rgbd -lopencv_saliency -lopencv_sfm -lopencv_stereo -lopencv_structured_light -lopencv_phase_unwrapping -lopencv_superres -lopencv_optflow -lopencv_surface_matching -lopencv_tracking -lopencv_datasets -lopencv_text -lopencv_dnn -lopencv_plot -lopencv_videostab -lopencv_videoio -lopencv_xfeatures2d -lopencv_shape -lopencv_ml -lopencv_ximgproc -lopencv_video -lopencv_xobjdetect -lopencv_objdetect -lopencv_calib3d -lopencv_imgcodecs -lopencv_features2d -lopencv_flann -lopencv_xphoto -lopencv_photo -lopencv_imgproc -lopencv_core
```


4. Xcode 에서 프로젝트 환경 설정

다시 Xcode의 Build Settings 검색창에 'other linker' 검색하여
'Linking → Other Linker Flags' 에 앞서 복사한 내용 입력

```
-I/usr/local/Cellar/opencv/4.2.0_1/include/opencv4/opencv -I/usr/local/Cellar/opencv/4.2.0_1/include/opencv4 -L/usr/local/Cellar/opencv/4.2.0_1/lib -lopencv_gapi -lopencv_stitching -lopencv_aruco -lopencv_bgsegm -lopencv_bioinspired -lopencv_ccalib -lopencv_dnn_objdetect -lopencv_dnn_superres -lopencv_dpm -lopencv_highgui -lopencv_face -lopencv_freetype -lopencv_fuzzy -lopencv_hfs -lopencv_img_hash -lopencv_line_descriptor -lopencv_quality -lopencv_reg -lopencv_rgbd -lopencv_saliency -lopencv_sfm -lopencv_stereo -lopencv_structured_light -lopencv_phase_unwrapping -lopencv_superres -lopencv_optflow -lopencv_surface_matching -lopencv_tracking -lopencv_datasets -lopencv_text -lopencv_dnn -lopencv_plot -lopencv_videostab -lopencv_videoio -lopencv_xfeatures2d -lopencv_shape -lopencv_ml -lopencv_ximgproc -lopencv_video -lopencv_xobjdetect -lopencv_objdetect -lopencv_calib3d -lopencv_imgcodecs -lopencv_features2d -lopencv_flann -lopencv_xphoto -lopencv_photo -lopencv_imgproc -lopencv_core
```



5. 예제 코드 실행

다음과 같은 예제 코드를 입력한 후, '**command + R**' 을 눌러 실행 결과 확인

데스크탑에 있는 'lena.png'라는 예시 이미지 파일을 출력하는 코드입니다.
예시 코드 작성해 보실 때, 이 부분은,
본인이 가지고 있는 이미지 파일을 경로와 함께 넣어주시면 됩니다.

```
#include <iostream>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>

using namespace cv;
using namespace std;

int main(int argc, const char * argv[]){
    Mat pic = imread("/Users/kimhyeyeong/Desktop/lena.png");
    if(pic.empty()){
        cerr << "read fail!" << endl;
        exit(-1);
    }
    imshow("Pic", pic);
    waitKey(0);

    return 0;
}
```

5. 예제 코드 실행 – 파일 상대 경로 사용

1) 프로젝트와 동일 폴더에 읽을 파일 넣은 경우

```
#include <iostream>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>

using namespace cv;
using namespace std;

int main(int argc, const char * argv[]){
    Mat pic = imread("lena.png");
    if(pic.empty()){
        cerr << "read fail!" << endl;
        exit(-1);
    }
    imshow("Pic", pic);
    waitKey(0);

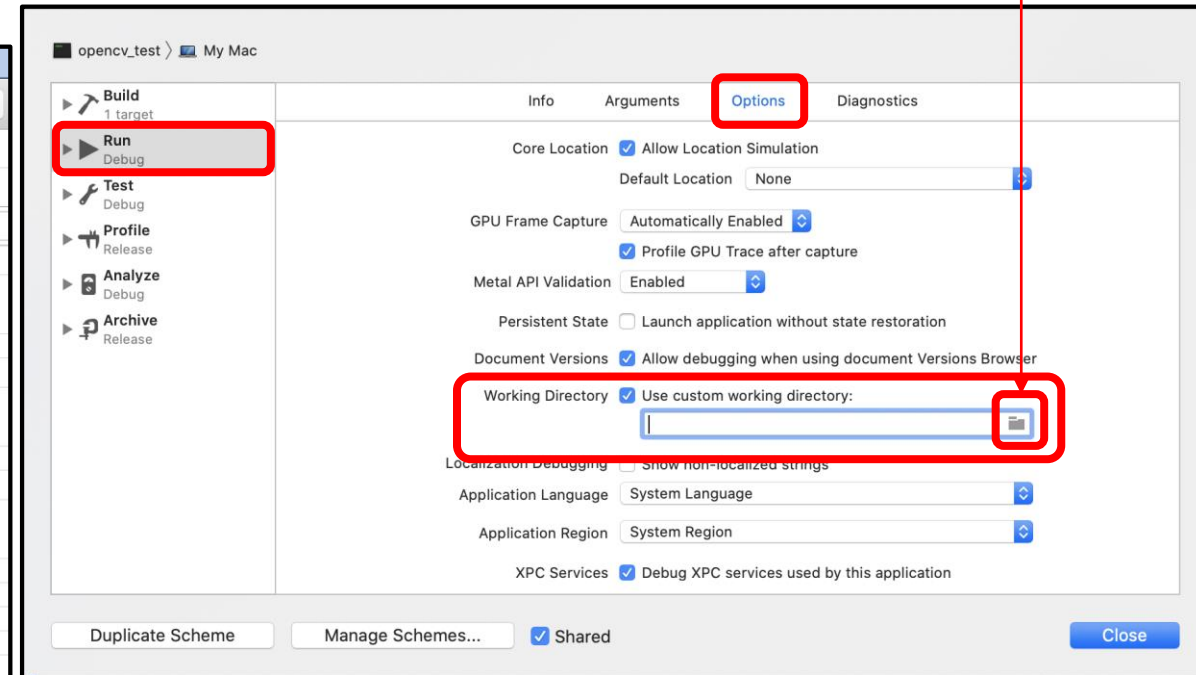
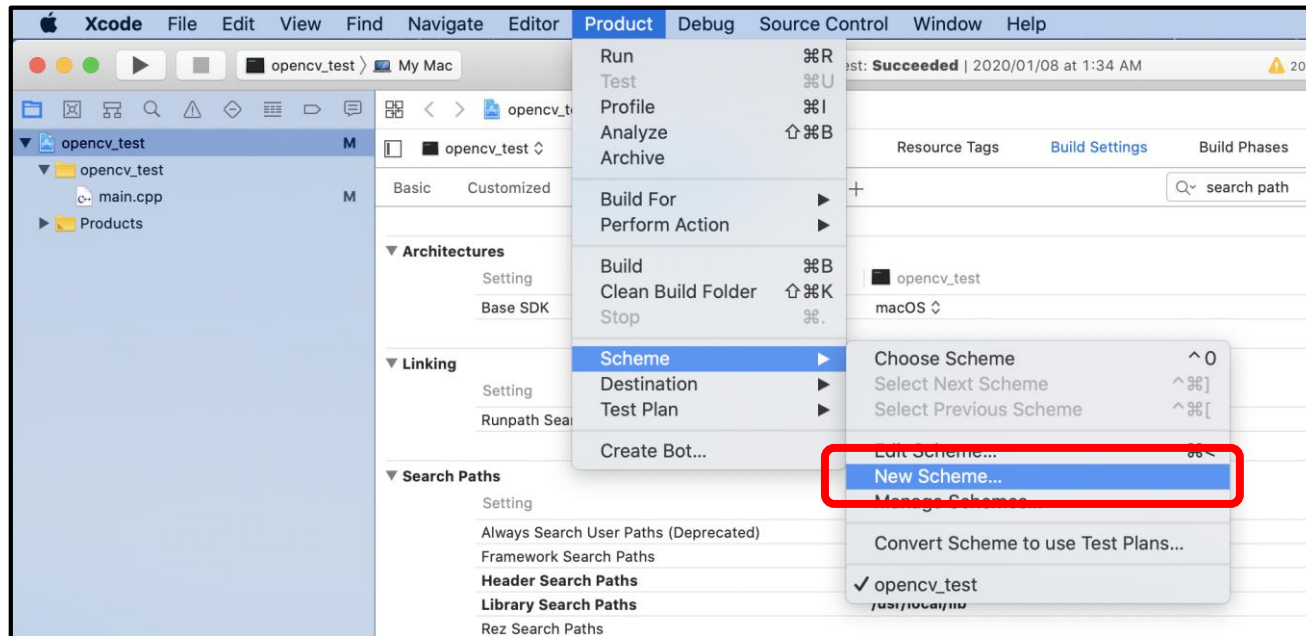
    return 0;
}
```

5. 예제 코드 실행 - 파일 상대 경로 사용

1) 프로젝트와 동일 폴더에 읽을 파일 넣은 경우

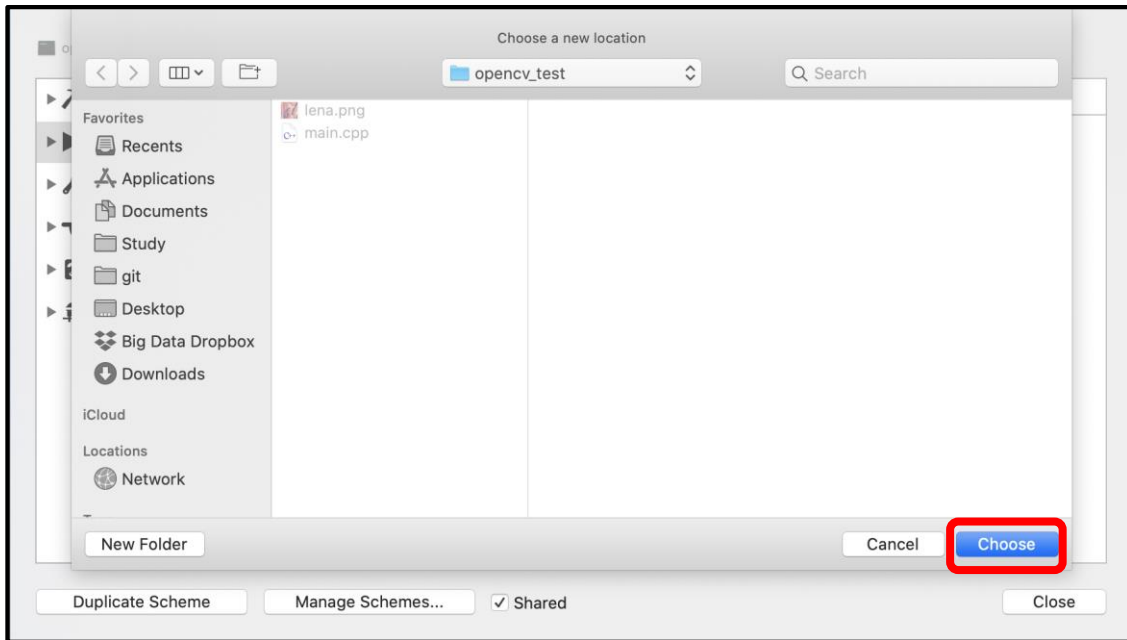
Product > Scheme > New Scheme

폴더 모양의 아이콘을 눌러서
프로젝트가 들어 있는 폴더를 선택

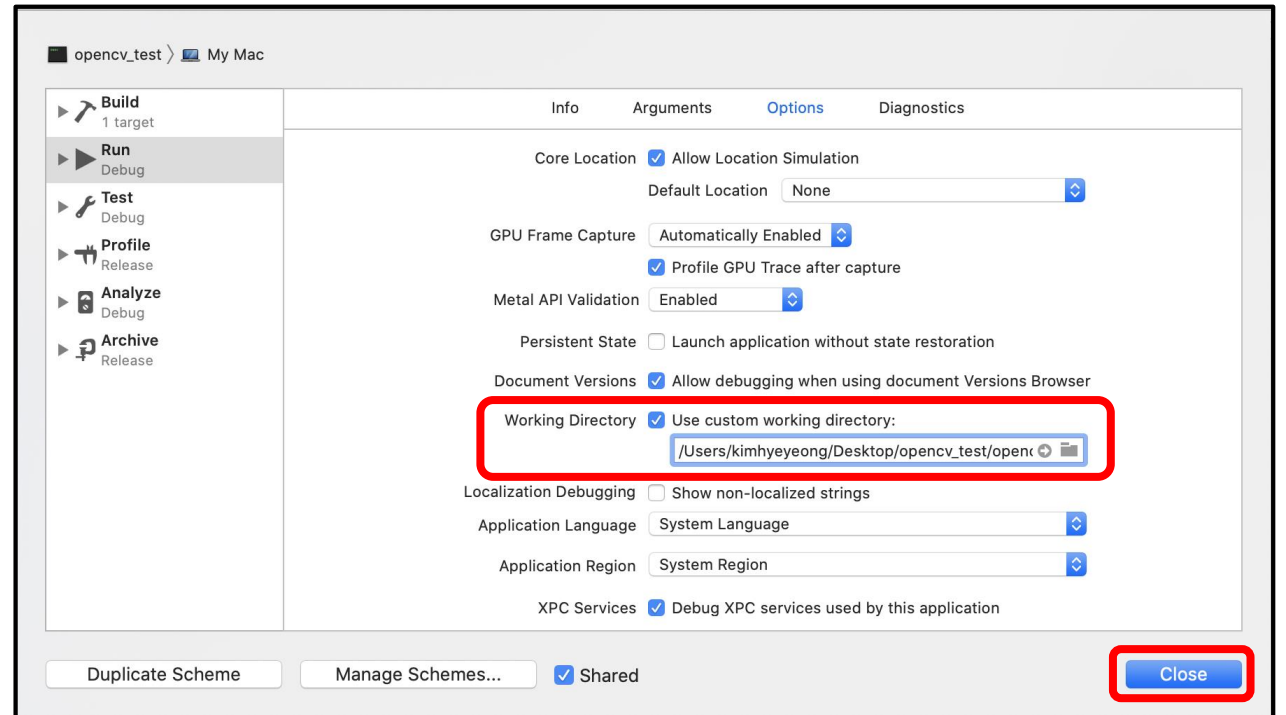


5. 예제 코드 실행 - 파일 상대 경로 사용

1) 프로젝트와 동일 폴더에 읽을 파일 넣은 경우



해당 프로젝트가 들어
있는 폴더를 선택



5. 예제 코드 실행 - 파일 절대 경로 사용

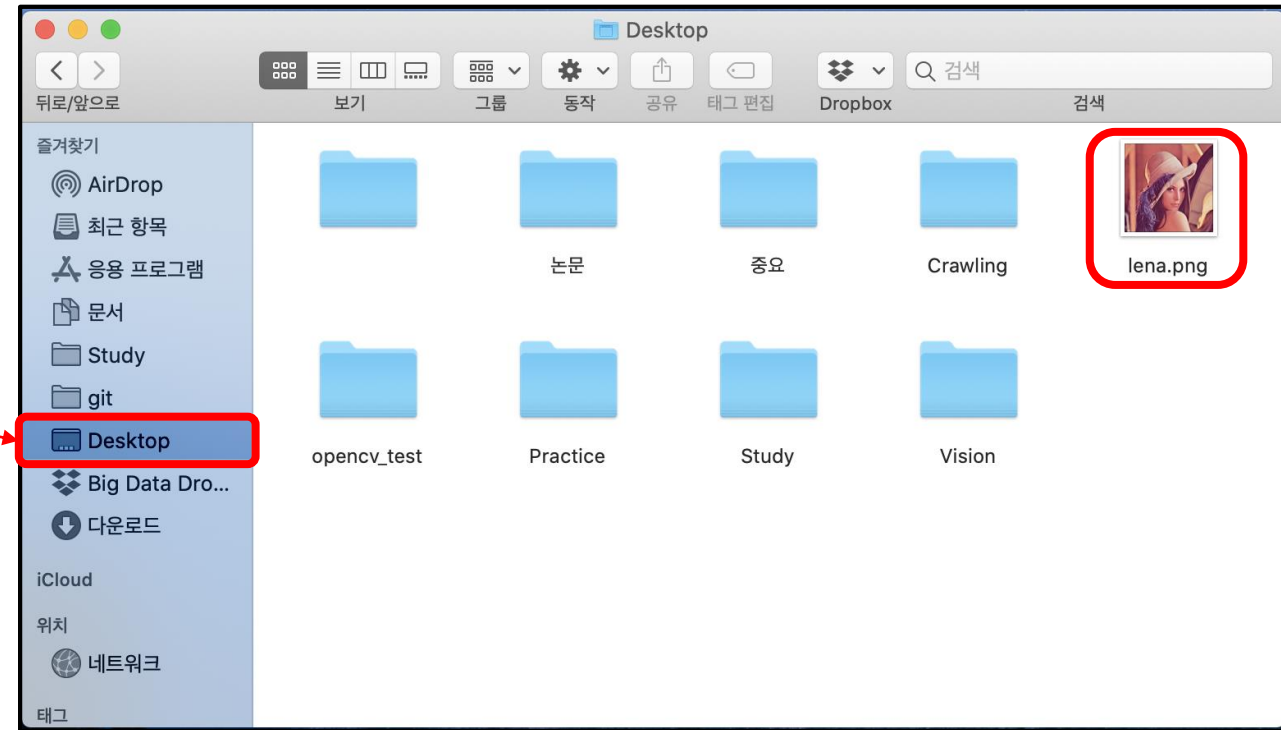
2) 파일의 절대 경로를 입력

```
#include <iostream>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>

using namespace cv;
using namespace std;

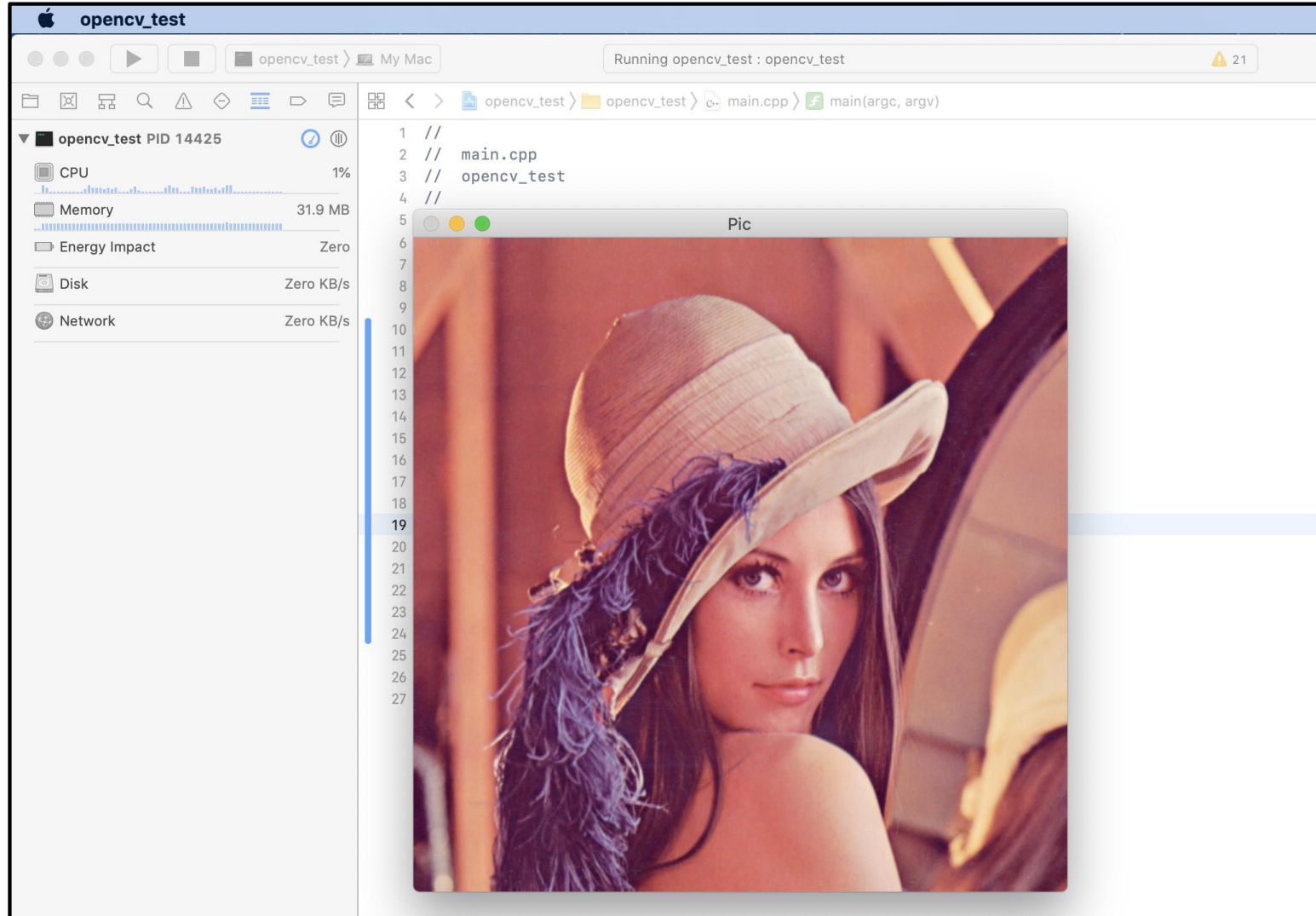
int main(int argc, const char * argv){
    Mat pic = imread("/Users/kimhyeyeong/Desktop/lena.png");
    if(pic.empty()){
        cerr << "read fail!" << endl;
        exit(-1);
    }
    imshow("Pic", pic);
    waitKey(0);

    return 0;
}
```



5. 예제 코드 실행 결과 확인

아래 화면과 같이, 이미지 파일이 정상적으로 출력되면 문제없이 설치가 완료된 것입니다.



Thank you!