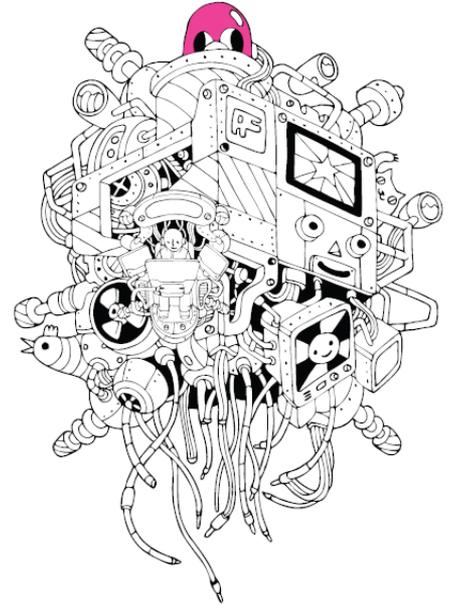


# 프로그래밍 언어 I



교재: 열혈강의 C 프로그래밍 개정판(윤성우 저)

Chapter 01. 이것이 C언어다.

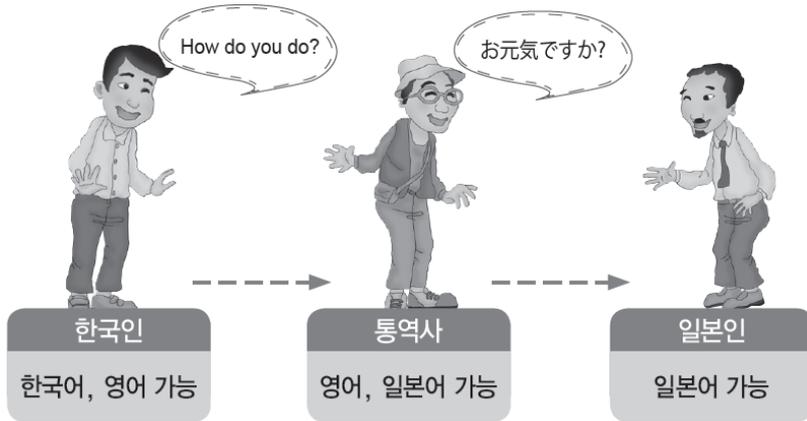
# 프로그래밍 언어 I



## Chapter 01-1. C언어의 개론적인 이야기

윤성우 저 열혈강의 C 프로그래밍 개정판

# C언어는 프로그래밍 언어이다.



√ 프로그래밍 언어란 무엇인가?

사람과 컴파일러가 이해할 수 있는 약속된 형태의 언어를 의미한다. C언어도 프로그래밍 언어 중 하나이다.

√ 컴파일러는?

프로그래밍 언어로 작성된 프로그램을 컴퓨터가 이해할 수 있는 기계어로 번역하는 역할을 한다.

√ 기계어(Machine Language)란?

컴퓨터가 이해할 수 있는 0과 1로 구성된 형태의 언어를 뜻함.

√ C언어를 공부한다는 것은?

문법을 이해하는 것.

표현능력을 향상시키는 것.

많이 사용할수록 표현에 능숙해진다.

다른 이의 표현을 참조할수록 표현이 부드러워진다.



# C언어의 역사와 특징

---

## √ C언어의 역사

1971년경 UNIX라는 운영체제의 개발을 위해 Dennis Ritchie와 Ken Thompson이 함께 설계한 범용적인 고급 (high-level)언어.

근원: ALGOL 60(1960) ▶ CPL(1963) ▶ BCPL(1969) ▶ B언어(1970)

## √ C언어 등장 이전의 유닉스 개발

어셈블리(assembly) 언어라는 저급(low-level)언어로 만들어졌었다.

그런데 어셈블리 언어는 하드웨어에 따라서 그 구성이 달라지기 때문에 CPU 별로 유닉스를 각각 개발해야만 했다.

## √ C언어 등장 이후 유닉스 개발

C언어의 구성은 CPU에 따라 나뉘지 않기 때문에 CPU별로 유닉스를 각각 개발할 필요가 없다.

## √ 고급언어? 저급언어?

사람이 이해하기 쉬운 언어는 고급언어, 기계어에 가까울 수록 저급언어.

C언어는 고급언어이면서 메모리에 직접 접근이 가능하기 때문에 저급언어의 특성도 함께 지닌다고 이야기 한다.

---



# C언어의 장점

---

√ C언어는 절차지향적 특성을 지닌다. 따라서 쉽게 익숙해질 수 있다.

인간의 사고하는 방식과 유사하다.

√ C언어로 작성된 프로그램은 이식성이 좋다.

CPU에 따라 프로그램을 재작성할 필요가 없다.

그러나 근래에는 C언어보다 이식성이 훨씬 뛰어난 언어들이 등장하고 있어서 장점으로 부각시키기에는 한계가 있다.

√ C언어로 구현된 프로그램은 좋은 성능을 보인다.

C언어를 이용하면 메모리의 사용량을 줄일 수 있고, 속도를 저하시키는 요소들을 최소화 할 수 있다.

단, 잘못 구현하면 오히려 성능이 좋지 못한 프로그램이 만들어지기도 한다.

**C언어의 장점은 앞으로 C언어를 공부해 나가면서 보다 정확히 이해하게 된다.**



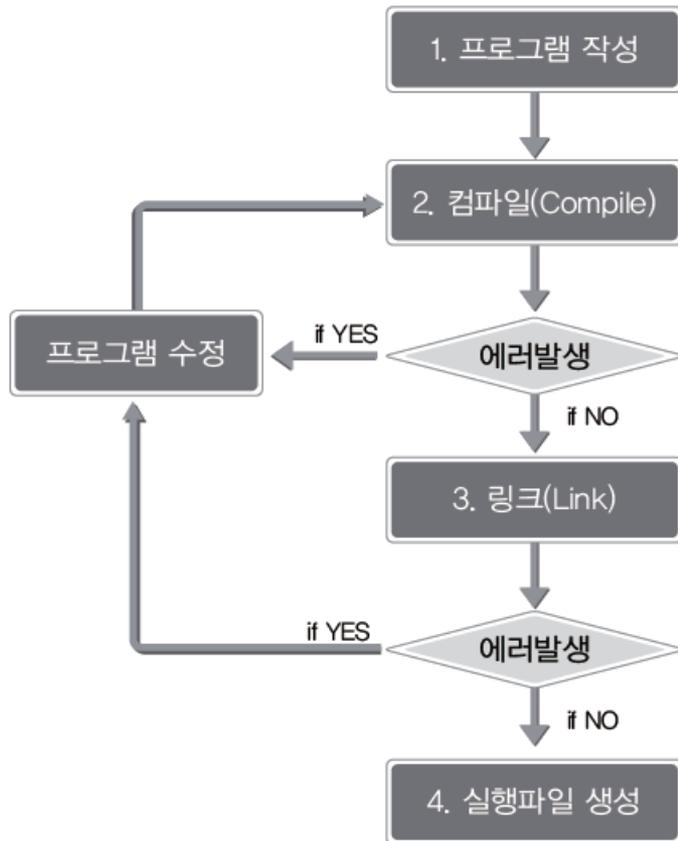
# 프로그래밍 언어 I



## Chapter 01-2. 프로그램의 완성과정

윤성우 저 열혈강의 C 프로그래밍 개정판

# C 프로그램 완성과정의 전체적인 이해



- 첫 번째 단계 : 프로그램의 작성
- 두 번째 단계 : 작성한 프로그램의 컴파일
- 세 번째 단계 : 컴파일 된 결과물의 링크

# 프로그래밍 언어 I

## Programming Language I

프로그래밍 도구의 설치

Google: [download visual studio community 2019](#)



# Visual Studio Community 2019 Download

<https://visualstudio.microsoft.com/ko/downloads/>

**다운로드** 선택 안내



**Visual Studio 2019**  
버전 16.8  
[릴리스 정보 >](#)

Android, iOS, Windows, 웹 및 클라우드에 대해 모든 기능을 갖춘 IDE(통합 개발 환경)

[버전 비교 >](#)  
[오프라인 설치 방법 >](#)

**커뮤니티**

강력한 IDE, 학생, 오픈 소스 제공자 및 개인을 위해 무료로 제공

**무료 다운로드** ↓

**Professional**

소규모 팀에 적합한 Professional IDE

**평가판** ↓

**Enterprise**

모든 규모의 팀을 위한 확장성 뛰어난 통합형 솔루션

**평가판** ↓

**Visual Studio Preview**  
[릴리스 정보 >](#)

주 릴리스에서는 아직 제공되지 않는 최신 기능에 미리 액세스  
[자세히 알아보기 >](#)



# Visual Studio Community 2019 설치 (1)

<https://visualstudio.microsoft.com/ko/thank-you-downloading-visual-studio/?sku=Community&rel=16>

## Visual Studio를 다운로드해 주셔서 감사합니다.

다운로드가 곧 시작됩니다. 다운로드가 시작되지 않은 경우에는 [여기를 클릭하여 다시 시도하세요](#)



### 지금 개발을 시작하세요.

- Visual Studio 설치에 대한 지원 받기
- Create your Desktop app today by following .NET (C#/F#/VB.NET) tutorial
- Get a jump start on Desktop app development with (C#/F#/VB.NET) quick start guide

컴퓨터

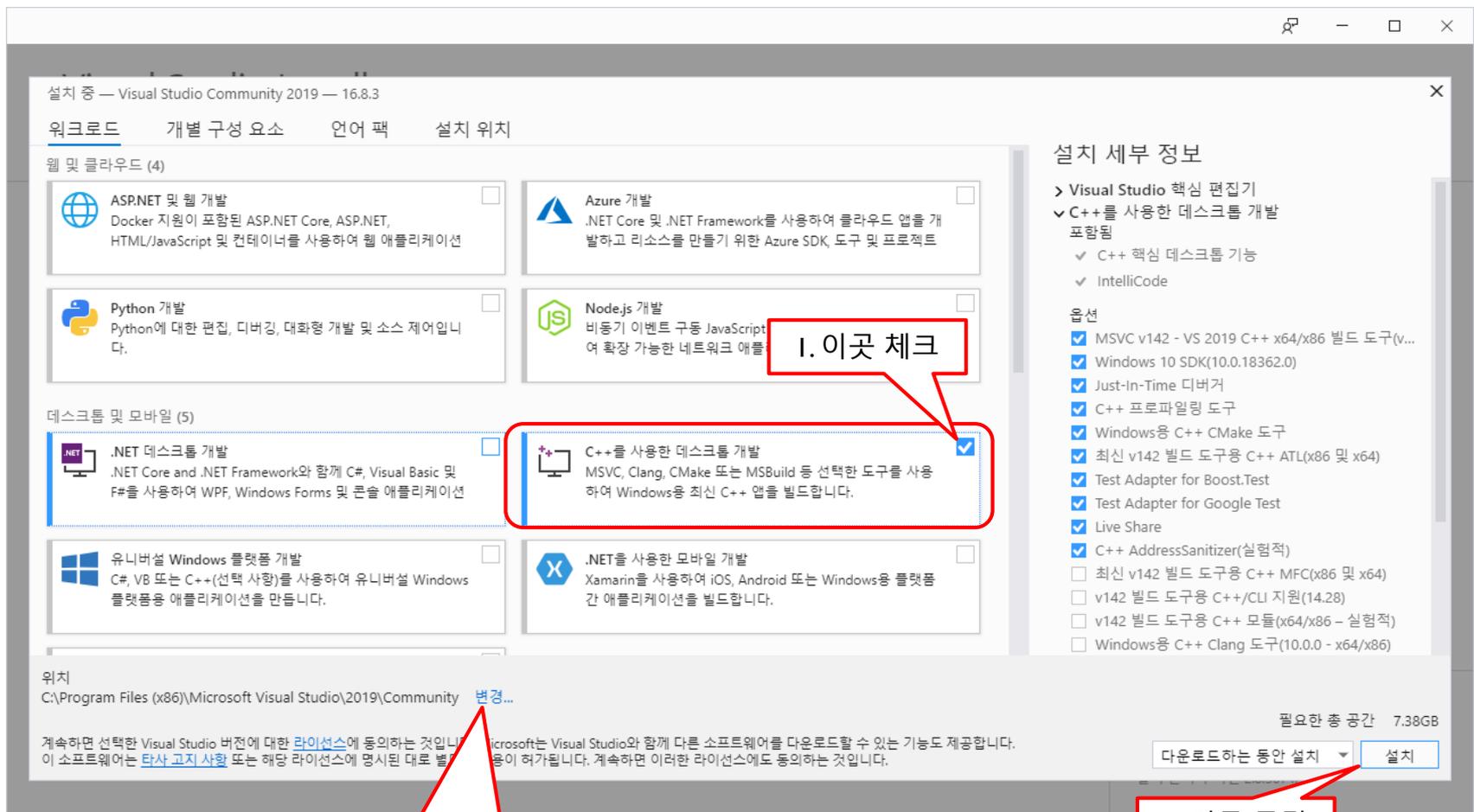
Visual Studio 둘러보기 >

vs\_community\_1...exe

click

모두 표시 x

# Visual Studio Community 2019 설치 (2)



2. 설치 위치를 바꾸려면  
이곳을 클릭해서 변경

3. 이곳 클릭

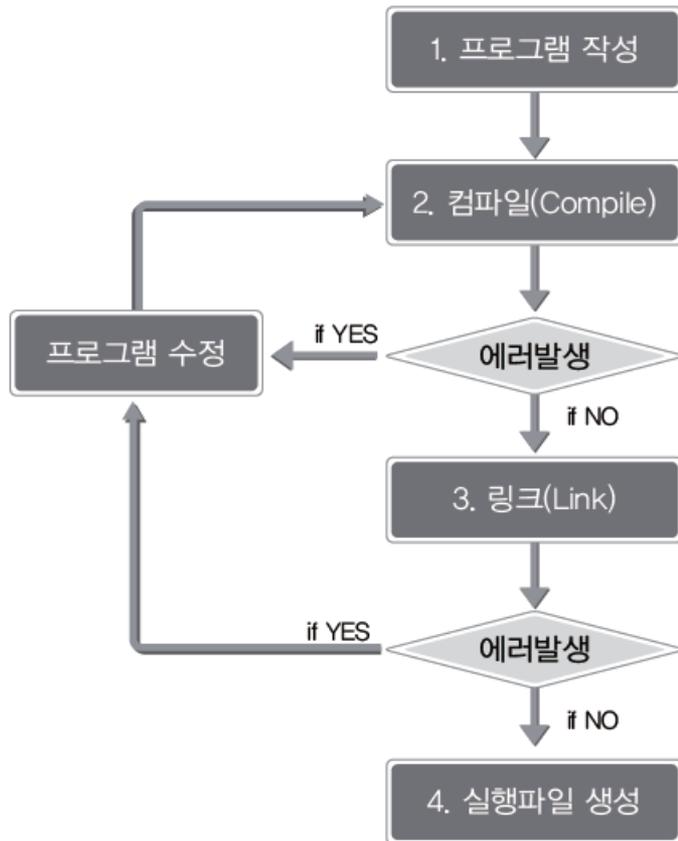
# 프로그래밍 언어 I

## Programming Language I

C 언어로 프로그램 만들어 보기



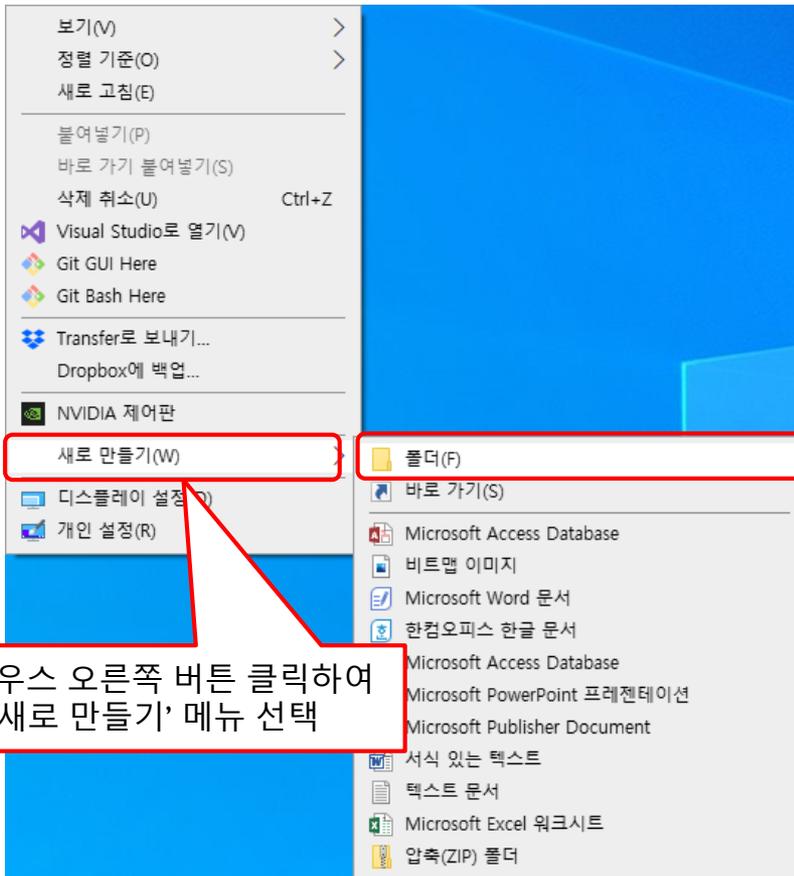
# C 프로그램 완성과정의 전체적인 이해



- 첫 번째 단계 : 프로그램의 작성
- 두 번째 단계 : 작성한 프로그램의 컴파일
- 세 번째 단계 : 컴파일 된 결과물의 링크

# Visual Studio Community 2019에서의 솔루션 생성1

「 바탕 화면 (혹은 자신이 원하는 곳)에 작업용 폴더 만들기 」



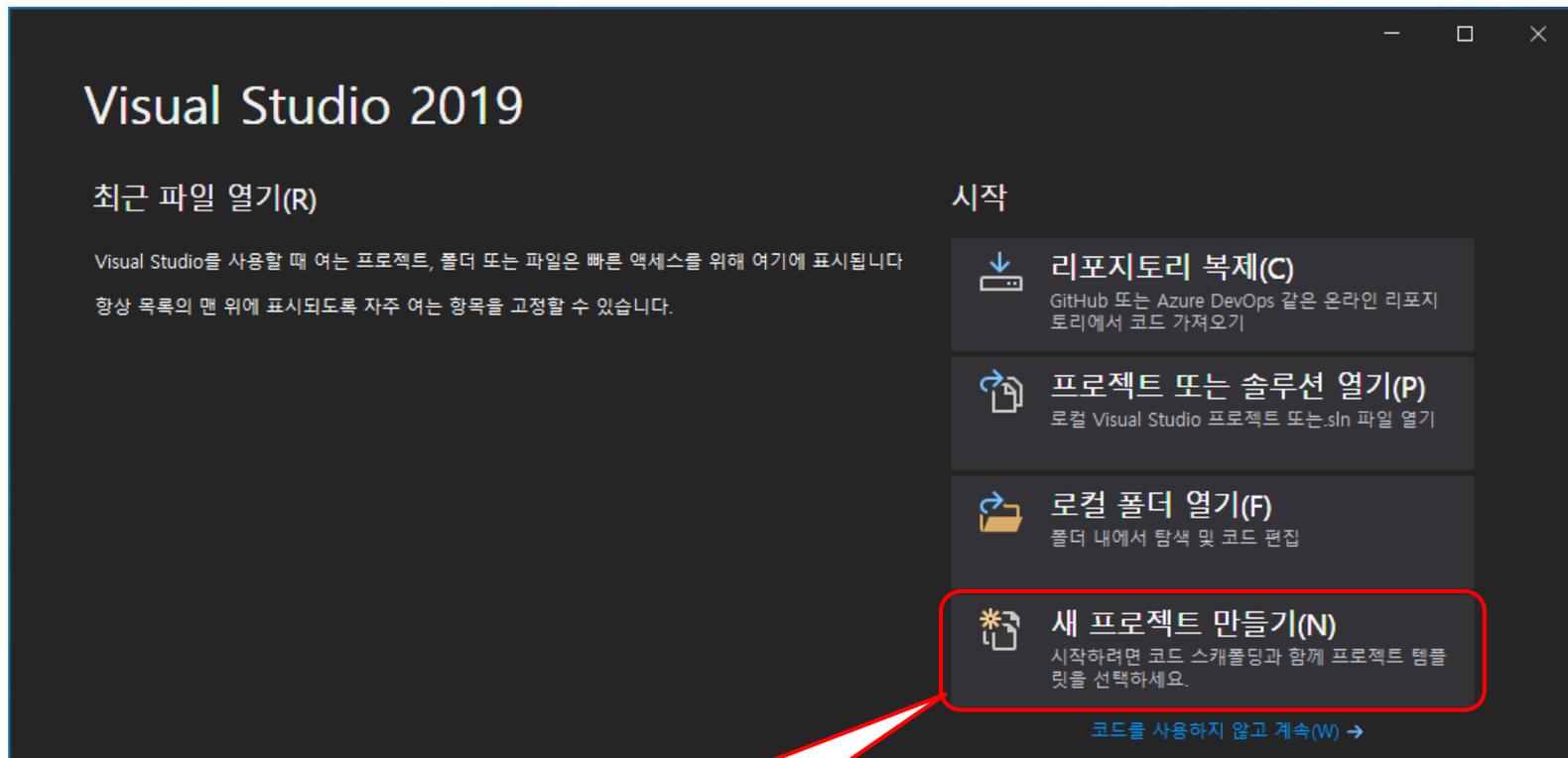
1. 바탕화면에서 마우스 오른쪽 버튼 클릭하여 팝업 메뉴 생성

3. '폴더' 메뉴 선택

4. 자신의 학번을 폴더명으로 입력

# Visual Studio Community 2019에서의 솔루션 생성2

## 「 Visual Studio Community 2019 Edition의 실행 」



‘새 프로젝트 만들기’ 선택

# Visual Studio Community 2019에서의 솔루션 생성3

## 「 Visual Studio Community 2019 Edition의 실행 」

새 프로젝트 만들기

최근 프로젝트 템플릿(R)

Windows 데스크톱 마법사 C++

템플릿 검색(Alt+S)(S)

모든 언어(L) 모든 플랫폼(P) 모든 프로젝트 형식(T)

빈 프로젝트  
Windows용 C++를 사용하여 처음부터 시작합니다. 시작 파일을 제공하지 않습니다.  
C++ Windows 콘솔

콘솔 앱  
Windows 터미널에서 코드를 실행합니다. 기본적으로 "Hello World"를 출력합니다.  
C++ Windows 콘솔

CMake 프로젝트  
.sln 또는 .vcxproj 파일에 종속되지 않은 최신 플랫폼 간 C++ 앱을 빌드하세요.  
C++ Windows Linux 콘솔

Windows 데스크톱 마법사  
마법사를 사용하여 고유한 Windows 앱을 만드세요.  
C++ Windows 데스크톱 콘솔 라이브러리

Windows 데스크톱 애플리케이션  
Windows에서 실행되는 그래픽 사용자 인터페이스를 사용하는 애플리케이션용 프로젝트입니다.  
C++ Windows 데스크톱

빈 솔루션

뒤로(B) 다음(N)

1. 'Windows 데스크톱 애플리케이션' 선택

2. '다음' 버튼 클릭

# Visual Studio Community 2019에서의 솔루션 생성4

## 「 Visual Studio Community 2019 Edition의 실행 」

새 프로젝트 만들기

최근 프로젝트 템플릿(R)

Windows 데스크톱 마법사 C++

템플릿 검색(Alt+S)(S)

모든 언어(L) 모든 플랫폼(P) 모든 프로젝트 형식(T)

빈 프로젝트  
Windows용 C++를 사용하여 처음부터 시작합니다. 시작 파일을 제공하지 않습니다.  
C++ Windows 콘솔

콘솔 앱  
Windows 터미널에서 코드를 실행합니다. 기본적으로 "Hello World"를 출력합니다.  
C++ Windows 콘솔

CMake 프로젝트  
.sln 또는 .vcxproj 파일에 종속되지 않은 최신 플랫폼 간 C++ 앱을 빌드하세요.  
C++ Windows Linux 콘솔

Windows 데스크톱 마법사  
마법사를 사용하여 고유한 Windows 앱을 만드세요.  
C++ Windows 데스크톱 콘솔 라이브러리

Windows 데스크톱 애플리케이션  
Windows에서 실행되는 그래픽 사용자 인터페이스를 사용하는 애플리케이션용 프로젝트입니다.  
C++ Windows 데스크톱

빈 솔루션

뒤로(B) 다음(N)

1. 'Windows 데스크톱 애플리케이션' 선택

2. '다음' 버튼 클릭

# Visual Studio Community 2019에서의 솔루션 생성5

「 프로그래밍을 위한 작업공간의 마련 」

The screenshot shows the '새 프로젝트 구성' (New Project Configuration) dialog in Visual Studio. The dialog is titled 'Windows 데스크톱 마법사' (Windows Desktop Wizard) and is set to 'C++' and 'Windows' project type. The project name is 'MyFirstProject'. The location is 'C:\Users\WIN\Desktop\20210225\'. The solution name is 'MyFirstProject'. There are four red callout boxes with white text and arrows pointing to specific elements: 1. '1. 프로젝트의 이름을 'MyFirstProject'로' (1. Set project name to 'MyFirstProject') pointing to the project name field. 2. '2. 이 버튼을 클릭' (2. Click this button) pointing to the ellipsis button next to the location field. 3. '3. 바탕화면에 만든 작업 폴더(학번) 지정' (3. Specify work folder (student ID) on desktop) pointing to the location field. 4. '4. '만들기' 버튼 클릭' (4. Click 'Create' button) pointing to the '만들기(C)' button at the bottom right.

새 프로젝트 구성

Windows 데스크톱 마법사 C++ Windows 데스크톱 콘솔 라이브러리

프로젝트 이름(N)  
MyFirstProject

1. 프로젝트의 이름을 'MyFirstProject'로

위치(L)  
C:\Users\WIN\Desktop\20210225\

2. 이 버튼을 클릭

3. 바탕화면에 만든 작업 폴더(학번) 지정

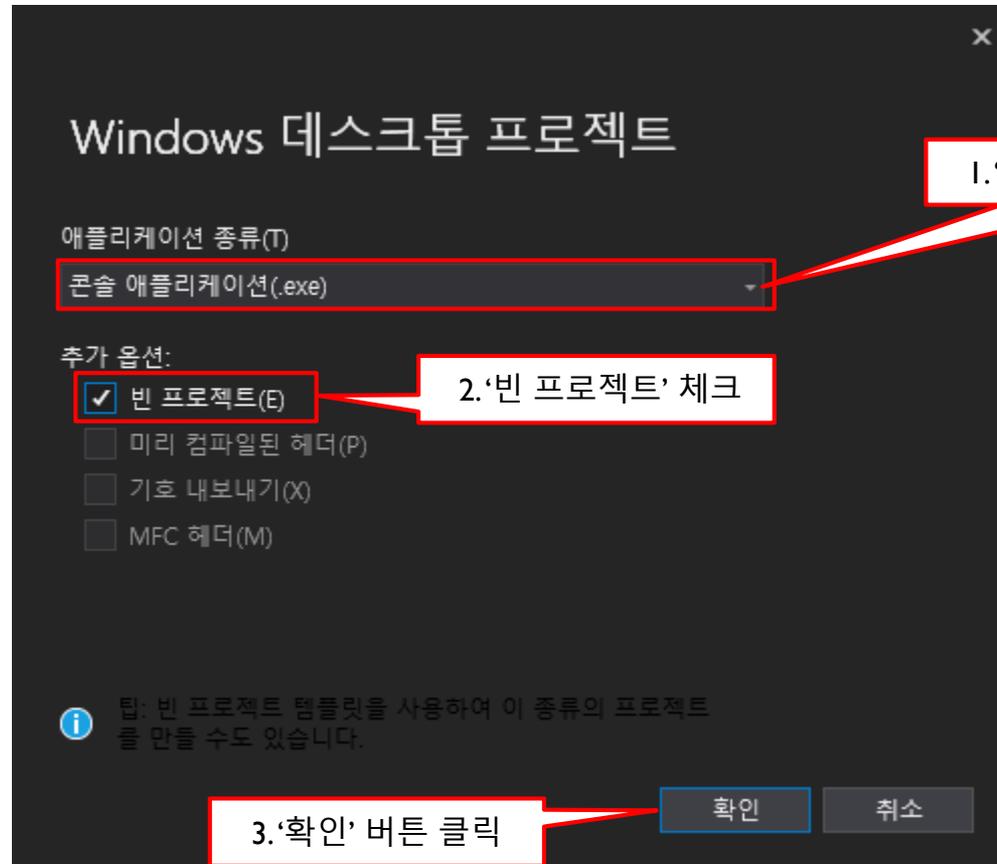
4. '만들기' 버튼 클릭

4. '만들기' 버튼 클릭

뒤로(B) 만들기(C)

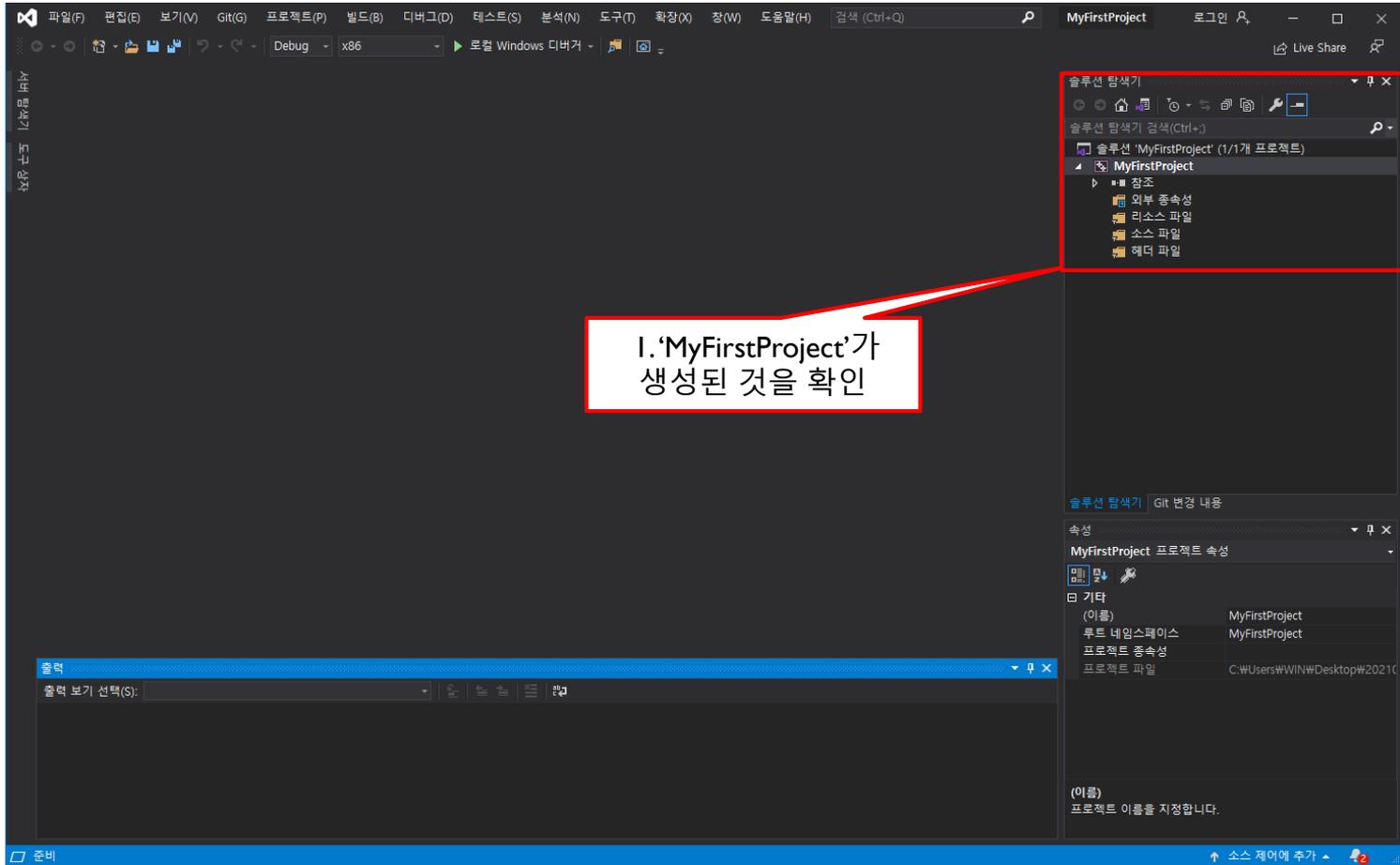
# Visual Studio Community 2019에서의 솔루션 생성6

## 「 프로젝트 생성을 위한 선택 」



# Visual Studio Community 2019에서의 솔루션 생성7

「 작업공간의 마련 완료 」 솔루션 탐색기에서 솔루션과 프로젝트가 생성되었음을 확인할 수 있다.



# 소스파일의 생성1

「 새로운 소스파일을 생성하여 프로젝트에 추가 」

솔루션 탐색기의 소스 파일 위치에서 마우스 오른쪽 버튼을 눌러서 추가

The image shows a screenshot of the Visual Studio Solution Explorer window. The '소스 파일' (Source Files) folder is selected, and a context menu is open. Three callouts provide instructions:

- 1. '소스 파일' 위에서 마우스 오른쪽 버튼 클릭 (Click the right mouse button over 'Source Files')
- 2. 마우스 커서를 '추가(D)' 메뉴 위로 이동 (Move the mouse cursor to the 'Add (D)' menu)
- 3. '새항목(W)' 메뉴를 마우스 왼쪽 버튼으로 클릭 (Click the 'New Item (W)' menu with the left mouse button)

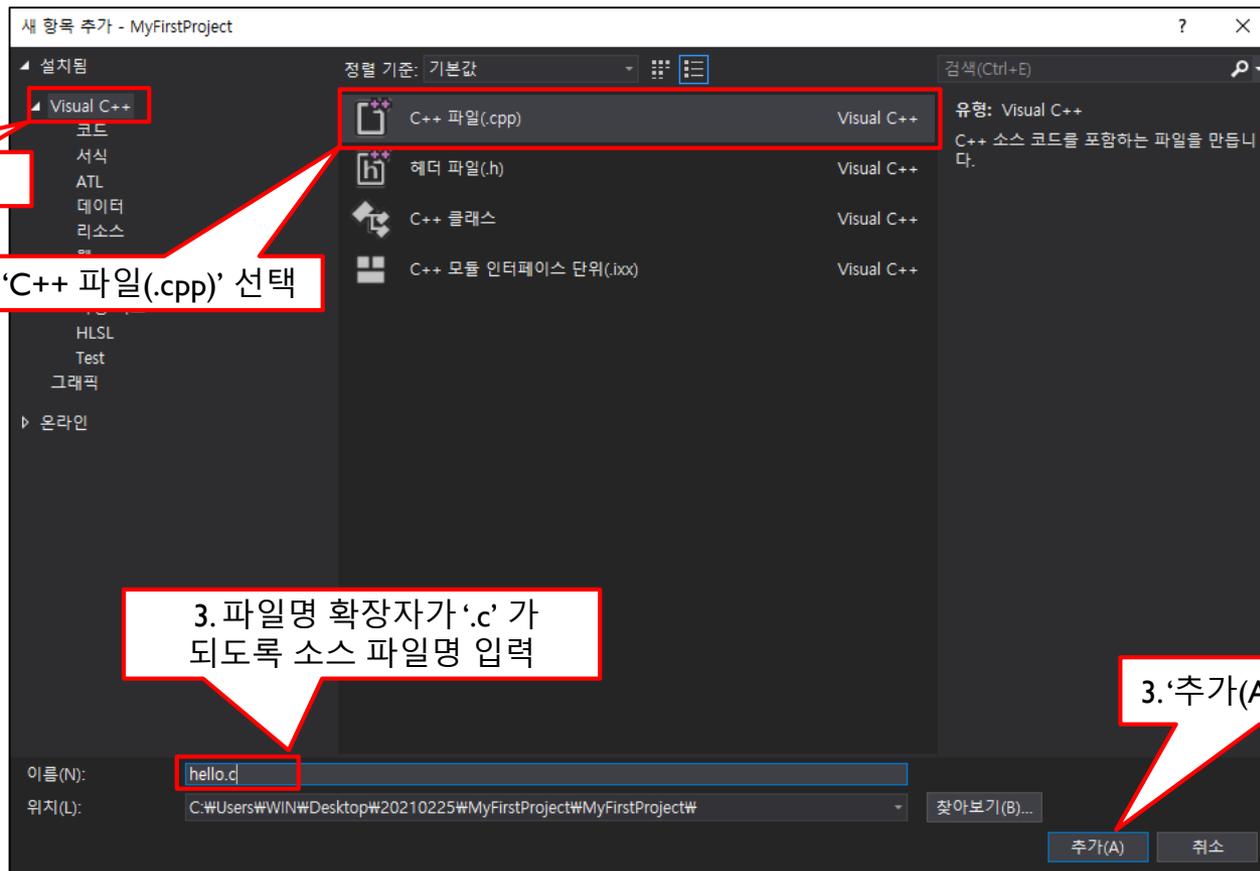
The context menu items include:

- 추가(D) (Add (D))
- 클래스 마법사(Z)... (Class Wizard (Z)...) - Ctrl+Shift+X
- 여기까지 범위 지정(S) (Specify range up to here (S))
- 새 솔루션 탐색기 뷰(N) (New Solution Explorer View (N))
- 잘라내기(T) (Cut (T)) - Ctrl+X
- 복사(Y) (Copy (Y)) - Ctrl+C
- 붙여넣기(P) (Paste (P)) - Ctrl+V
- 삭제(D) (Delete (D)) - Del
- 이름 바꾸기(M) (Rename (M)) - F2
- Open in Terminal
- 속성(R) (Properties (R)) - Alt+Enter
- 새 항목(W)... (New Item (W)...) - Ctrl+Shift+A
- 기존 항목(G)... (Existing Item (G)...) - Shift+Alt+A
- 새 필터(F) (New Filter (F))
- 모듈(M)... (Module (M)...) - Ctrl+M
- 클래스(C)... (Class (C)...) - Ctrl+C
- 리소스(R)... (Resource (R)...) - Ctrl+R
- 새 EditorConfig (New EditorConfig)

# 소스파일의 생성2

「 확장자가 .c인 소스파일 생성 」

확장자에 따라서 컴파일의 형식을 결정! 확장자를 .c로 해야 C언어 기반으로 컴파일 한다.



1. 'Visual C++' 선택

2. 'C++ 파일(.cpp)' 선택

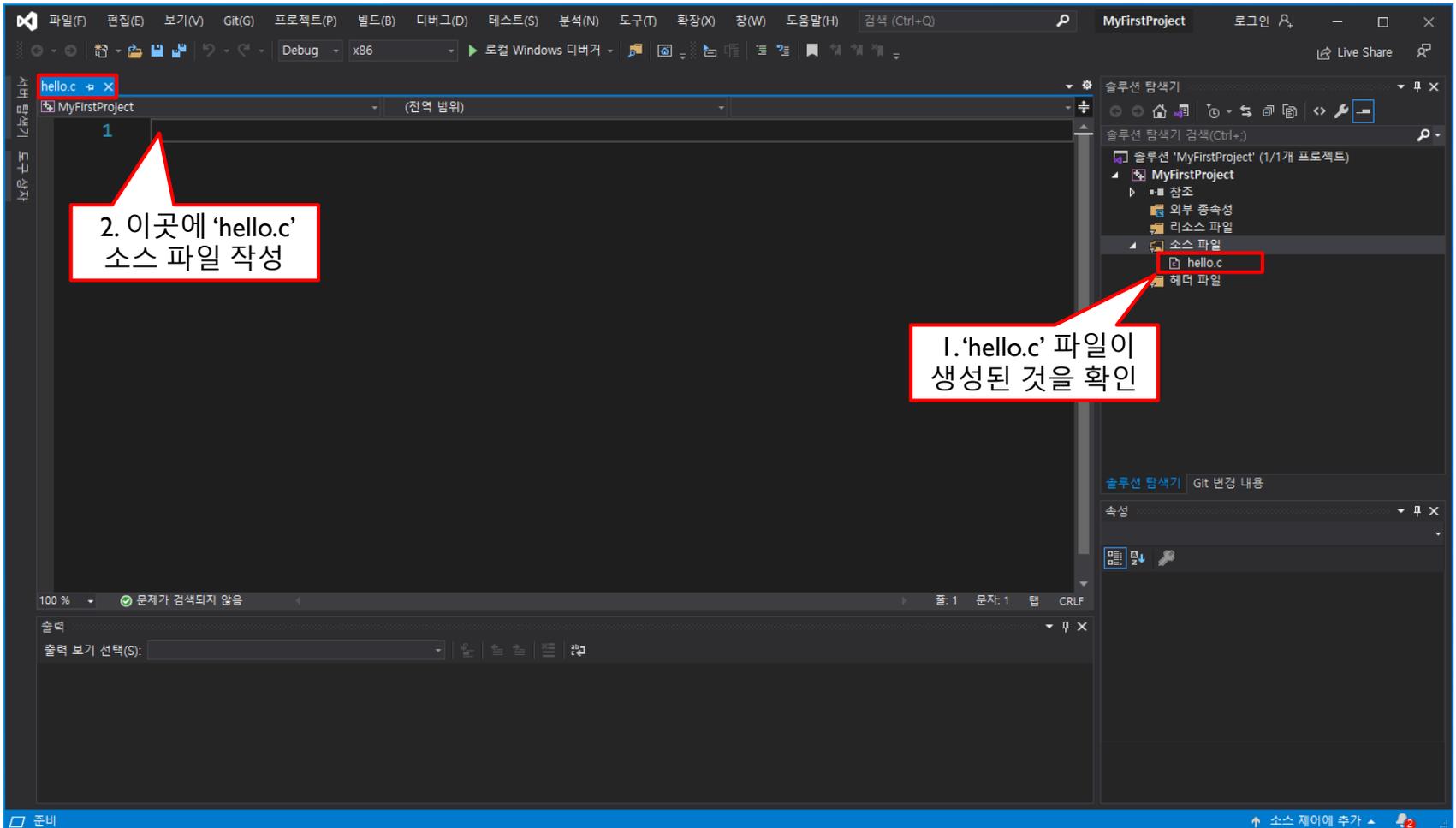
3. 파일명 확장자가 '.c' 가 되도록 소스 파일명 입력

3. '추가(A)' 버튼 클릭



# 생성된 소스파일

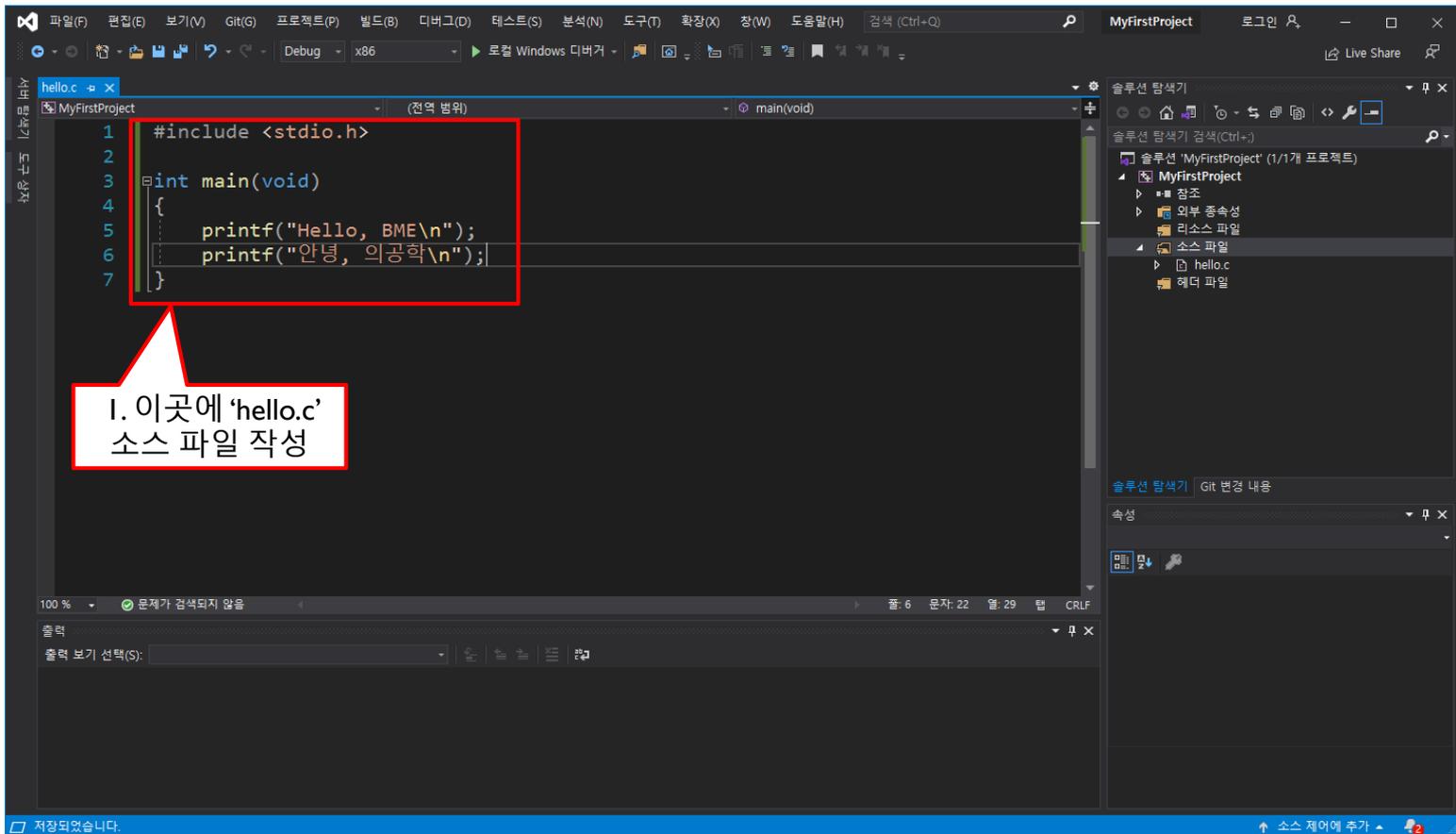
## 「 소스파일 생성 완료 」



# 생성된 소스파일과 프로그램의 입력

## 「 소스코드의 입력 」

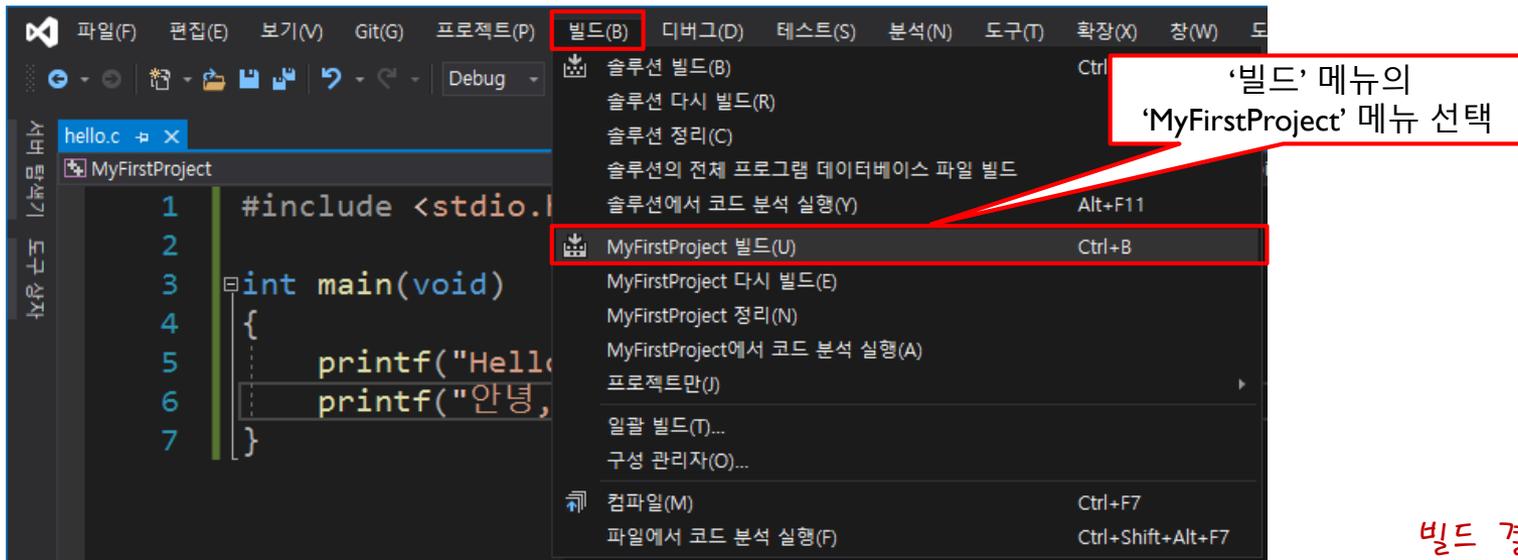
생성된 소스파일에 소스코드를 입력한다.



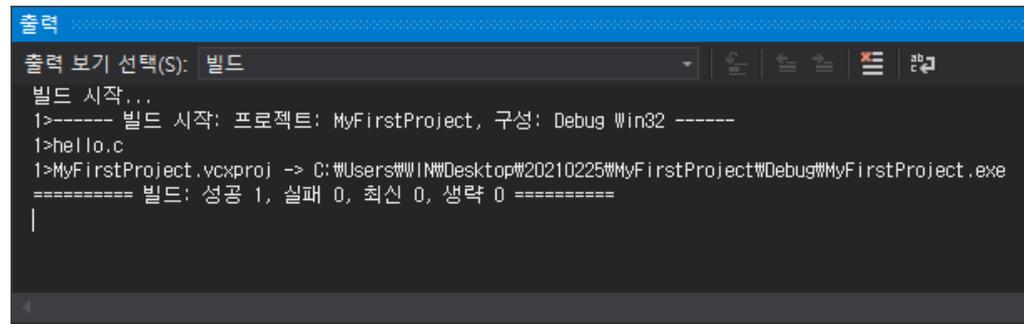
# 컴파일 및 실행결과 확인

「 컴파일과 실행결과 확인 」

소스파일을 빌드(컴파일/링크)한다.

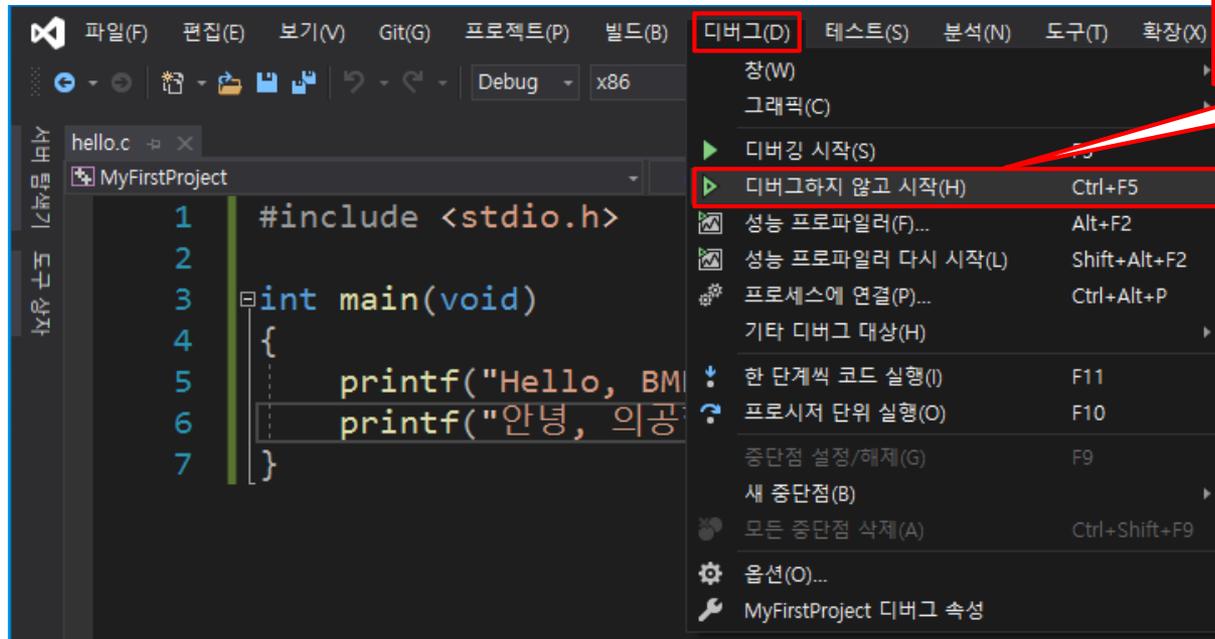


빌드 결과

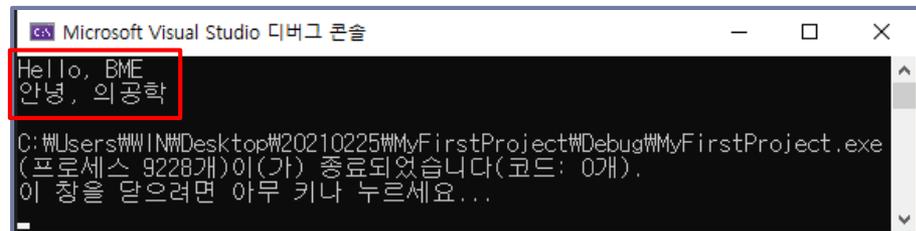


# 컴파일 및 실행결과 확인

「 컴파일과 실행결과 확인 」 빌드된 프로그램을 실행시킨다.



실행결과





Chapter **이**이 끝났습니다. 질문 있으신지요?