

# 심화프로그래밍

실습강의 1주차

# 실습강의 소개

## ● 실습 진행 방법

- 간단한 이론 복습 및 해당주차 실습과제 설명
- 실습 후 보고서와 소스코드를 압축하여 **수요일 자정(23:59)까지**  
꼭!! 이클래스 제출(**이메일 제출 불가, 반드시 이클래스를 통해 제출**)
- 실습과제 제출기한 엄수(제출기한 이후로는 0점 처리)
- **보고서는 출력하여 수업 시작 전에 제출(대면으로 실습수업 진행 시)**

## ● Q & A

- 이클래스 및 실습조교 이메일을 통해 질의응답
- **이메일 제목 : [심화프로그래밍\_홍길동] \*본인 과목명과 성명 꼭 작성!!**
- 실습조교 메일 주소 : [0hae@dgu.ac.kr](mailto:0hae@dgu.ac.kr), [wundermilch@dgu.ac.kr](mailto:wundermilch@dgu.ac.kr)

# 실습 보고서 작성 방법 [1/2]

- 실습 보고서 (10 페이지 이내)

- 문제 분석: 실습 문제에 대한 요구 사항 파악, 해결 방법 등 기술

- 프로그램 설계 및 알고리즘

- 해결 방법에 따라 프로그램 설계 및 알고리즘 등 기술
- e.g.) 문제 해결 과정 및 핵심 알고리즘 기술

- 소스코드 및 주석

- 소스코드와 그에 해당하는 주석 첨부
- 각각의 함수가 수행하는 작업, 매개변수, 반환 값 등을 명시
- 소스코드 전체 첨부 (소스코드 화면 캡처X, 소스코드는 복사/붙여넣기로 첨부)

- 결과 및 결과 분석

- 결과 화면을 캡처 하여 첨부, 해당 결과가 도출된 이유와 타당성 분석

- 소감

- 실습 문제를 통해 습득할 수 있었던 지식, 느낀 점 등을 기술

# 실습 보고서 작성 방법 [2/2]

## ● 제출 방법

- 보고서, 소스코드, 실행파일을 1개의 파일로 압축하여 e-class "과제" 메뉴를 통해 제출
  - "이름학번실습주차.zip" 형태로 제출(e.g. :김동국19919876실습1.zip)
  - 파일명에 공백, 특수 문자 등 사용 금지
- 대면 수업일 경우 출력한 보고서를 실습 시간에 제출

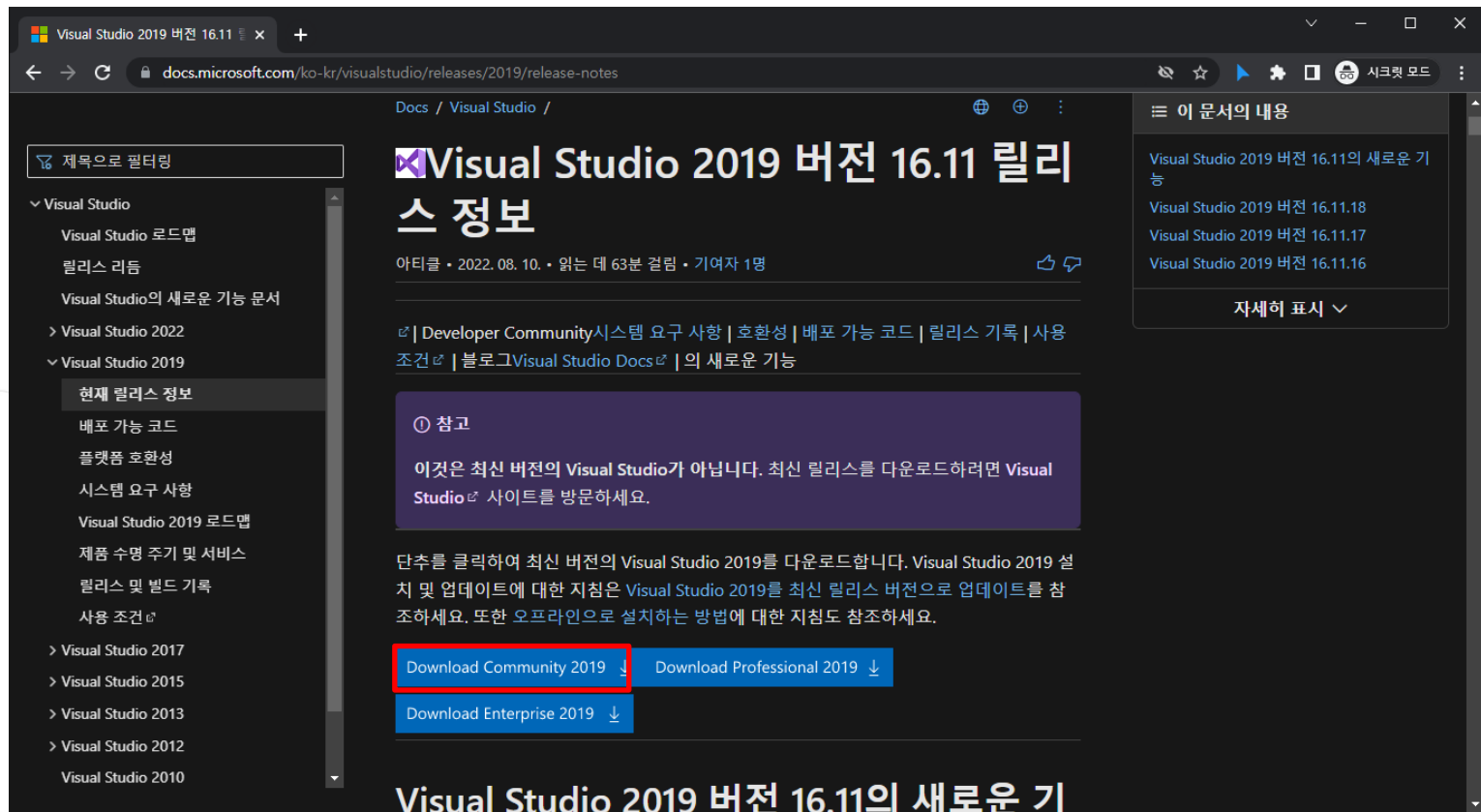
## ● 유의 사항

- 보고서의 표지에는 학과, 학번, 이름, 담당 교수님, 제출일자 반드시 작성
- 정해진 기한내 제출
  - 기한 넘기면 **0점** 처리
  - 이클래스가 과제 제출 마지막 날 오류로 동작하지 않을 수 있으므로, 최소 1~2일전에 제출
  - **과제 제출 당일 이클래스 오류로 인한 미제출은 불인정**
- 소스코드, 보고서를 자신이 작성하지 않은 경우 **실습 전체 점수 0점 처리**
- **Visual Studio 2019 또는 Sharstra 웹 IDE 기반 학습 프로그램 사용하여 실습 진행**

# Visual Studio 2019 설치 [1/3]

## ● Visual Studio 2019 다운로드

- 홈페이지(<https://docs.microsoft.com/ko-kr/visualstudio/releases/2019/release-notes>) 접속 → Download Community 2019 클릭



# Visual Studio 2019 설치 [2/3]

## ● Visual Studio 2019 설치

- 데스크톱 및 모바일 "C++를 사용한 데스크톱 개발 체크" → 기타 도구 집합 "Visual Studio 확장 개발" 체크 → 다운로드하는 동안 설치 → Install 클릭

### 데스크톱 및 모바일 (5)



#### .NET을 사용한 모바일 개발

Xamarin을 사용하여 iOS, Android 또는 Windows용 플랫폼 간 애플리케이션을 빌드합니다. 여기에는 선택적 설치로 ....



#### C++를 사용한 데스크톱 개발

MSVC, Clang, CMake 또는 MSBuild 등 선택한 도구를 사용하여 Windows용 최신 C++ 앱을 빌드합니다.



#### C++를 사용한 모바일 개발

C++를 사용하여 iOS, Android 또는 Windows용 플랫폼 간 애플리케이션을 빌드합니다.



### 기타 도구 집합 (5)



#### 데이터 스토리지 및 처리

SQL Server, Azure Data Lake 또는 Hadoop을 사용하여 데이터 솔루션을 연결, 개발 및 테스트하세요.



#### Visual Studio 확장 개발

Visual Studio용 추가 기능 및 확장(새 명령, 코드 분석기 및 도구 창 포함)을 만듭니다.



#### C++를 사용한 Linux 및 임베디드 개발

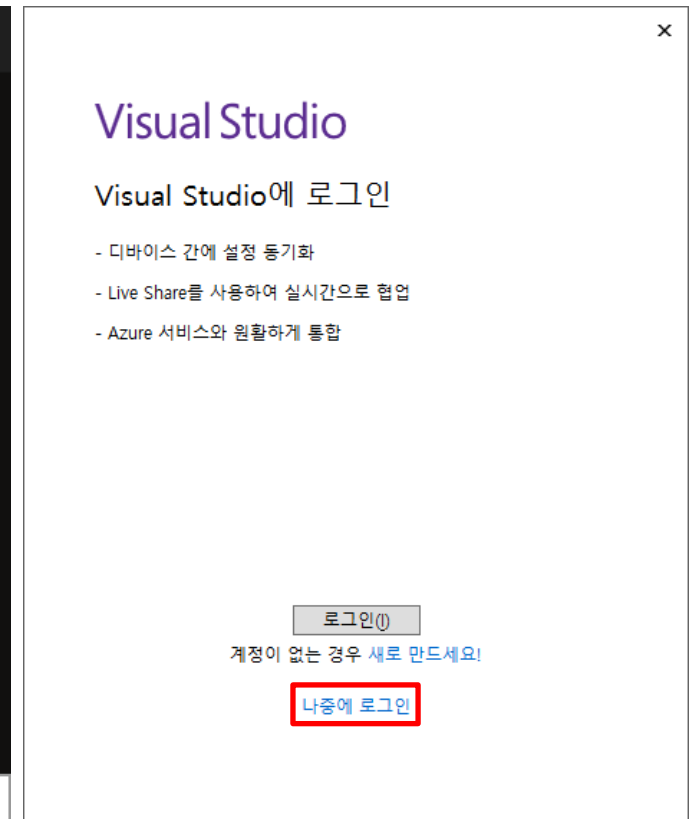
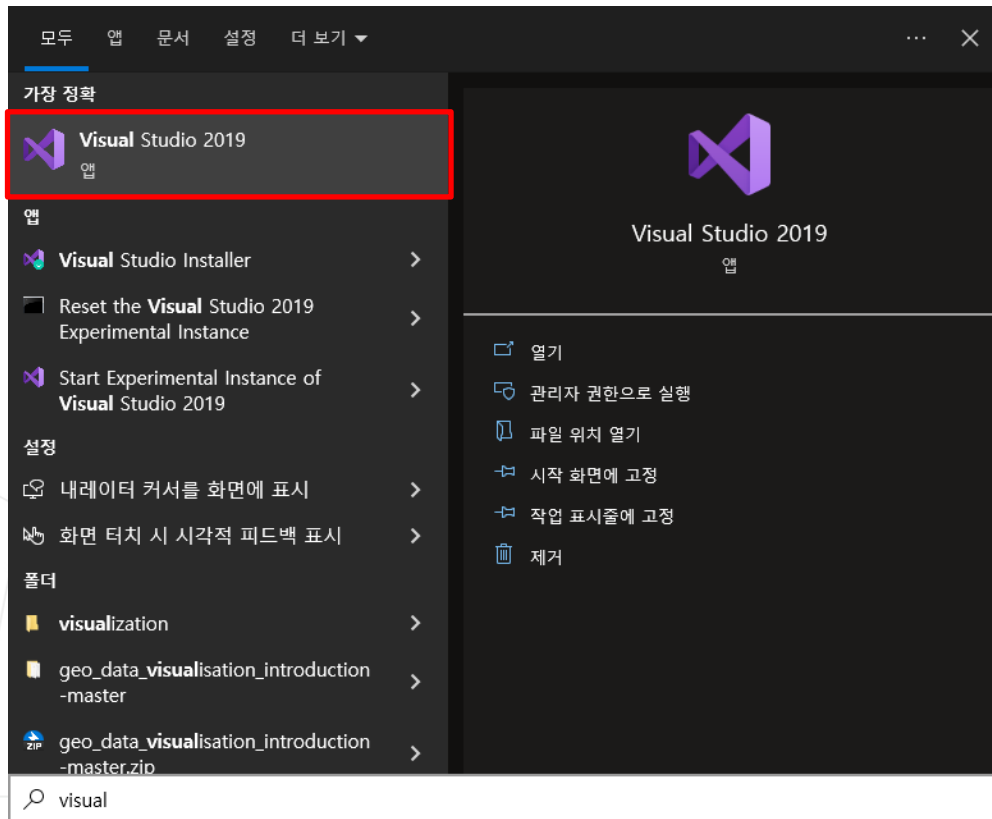
Linux 환경 또는 임베디드 디바이스에서 실행 중인 애플리케이션을 만들고 디버그합니다.



# Visual Studio 2019 설치 [3/3]

## ● Visual Studio 2019 설치

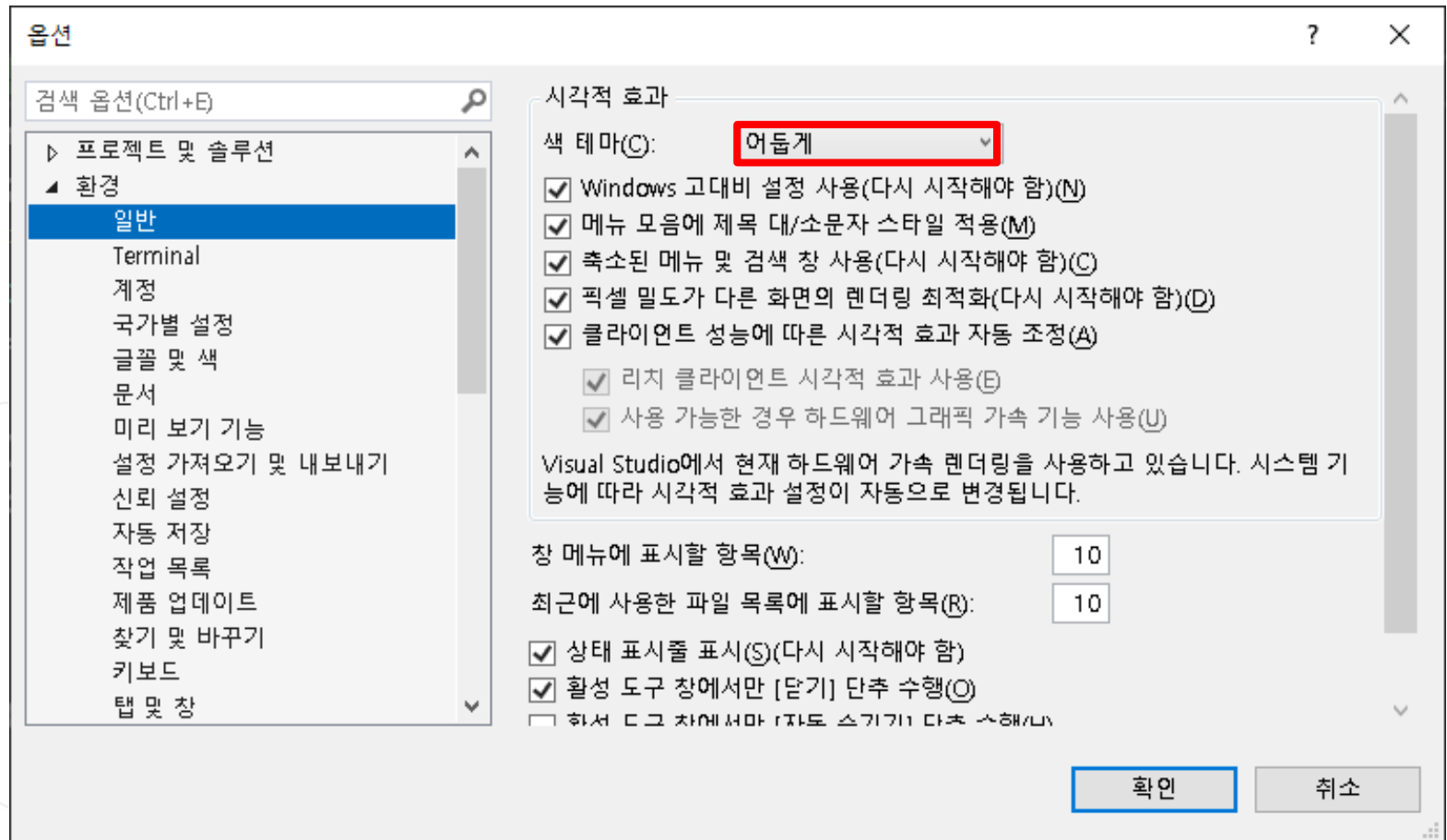
- 설치 완료 후 윈도우 시작 버튼을 통해 **Visual Studio 2019** 검색 후 프로그램 실행
- **Start Experimental Instance of Visual Studio 2019 프로그램 실행 주의**



# Visual Studio 2019 사용법 [1/13]

## ● Visual Studio 2019 개발환경 구성

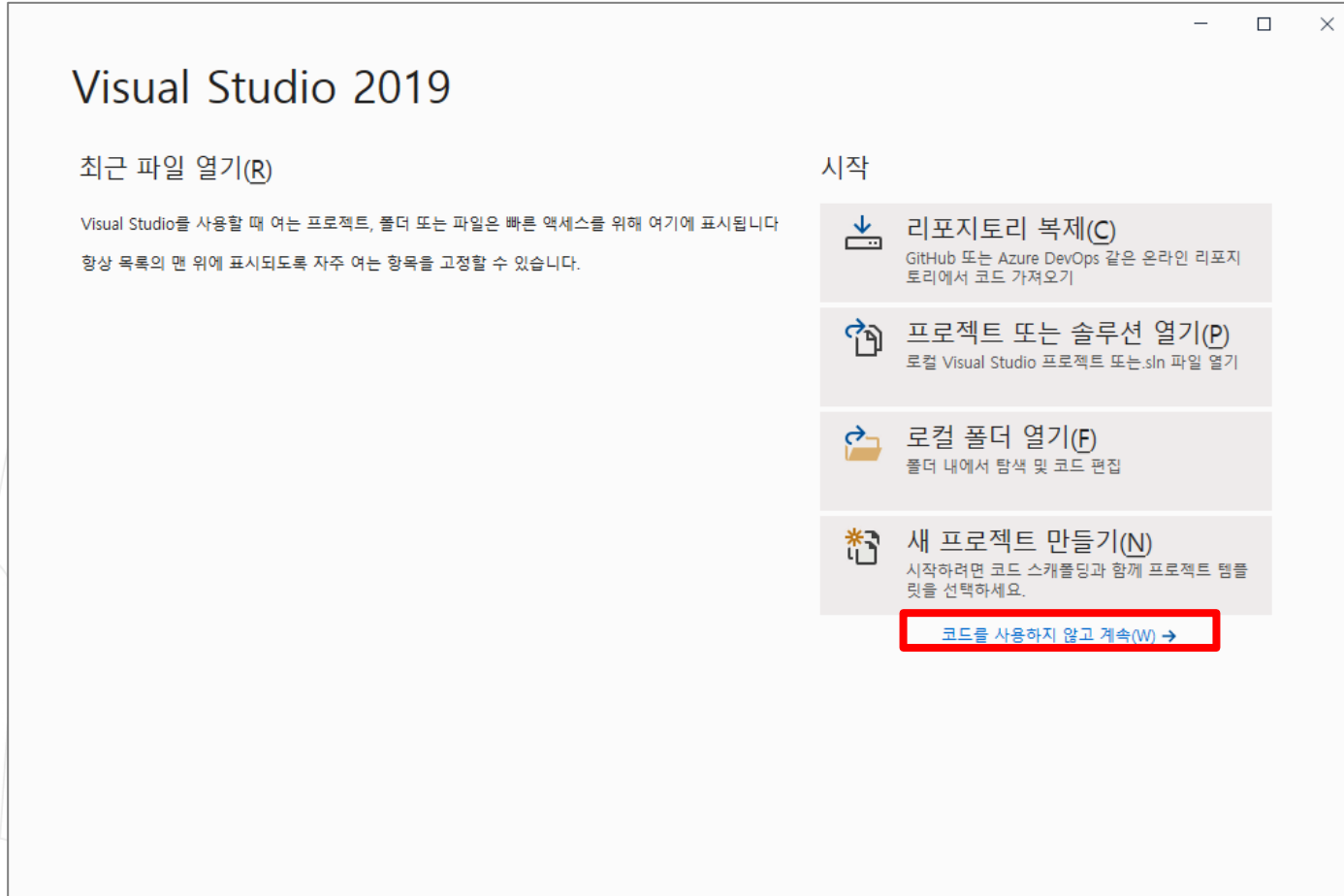
- 파일 → 도구 → 옵션 → 좌측 "환경" → 색 테마 "밝게 or 어둡게" (선택사항)





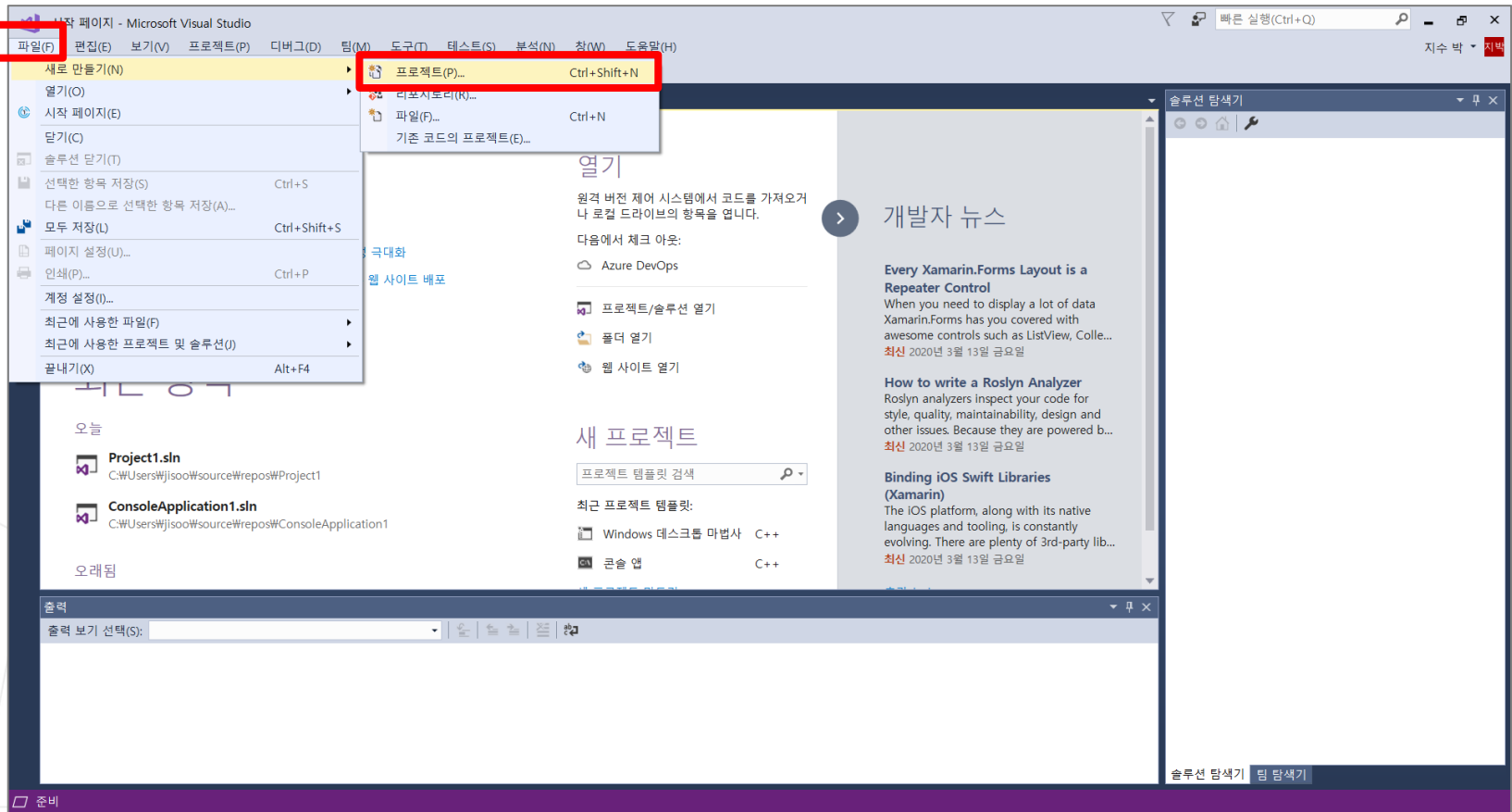
# Visual Studio 2019 사용법 [2/13]

- Visual Studio 2019 개발환경 구성
  - 코드를 사용하지 않고 계속(W)을 클릭하여 프로그램 실행



# Visual Studio 2019 사용법 [3/13]

- Visual Studio 2019 프로젝트 만들기
  - 파일 → 새로 만들기(N) → 프로젝트(P)



# Visual Studio 2019 사용법 [4/13]

- Visual Studio 2019 프로젝트 만들기
  - 새 프로젝트 만들기 → Windows 데스크톱 마법사 → 다음(N)



## ● Visual Studio 2019 프로젝트 만들기

- 프로젝트 이름(I) 작성 → 프로젝트가 저장되는 위치 확인 → 솔루션 및 프로젝트를 같은 디렉토리에 배치(D) 체크 → 만들기(C)

새 프로젝트 구성

Windows 데스크톱 마법사 C++ Windows 데스크톱 콘솔 라이브러리

프로젝트 이름(I)

심화프로그래밍\_20220907\_1주차

위치(L)

C:\Users\#twoimo\source\repos

솔루션 이름(M) ⓘ

심화프로그래밍\_20220907\_1주차

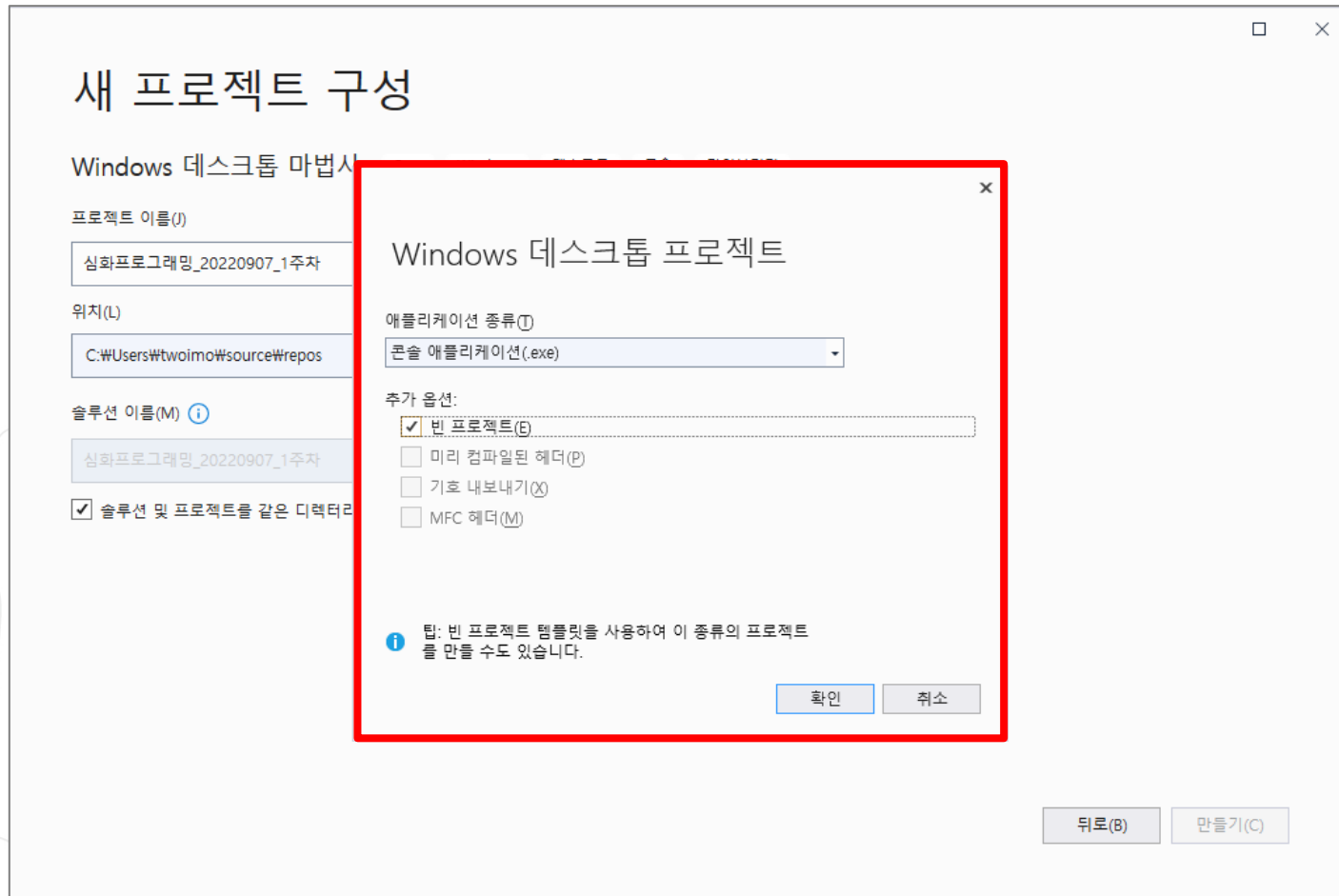
☒ 솔루션 및 프로젝트를 같은 디렉토리에 배치(D)

뒤로(B) 만들기(C)

# Visual Studio 2019 사용법 [6/13]

## ● Visual Studio 2019 프로젝트 만들기

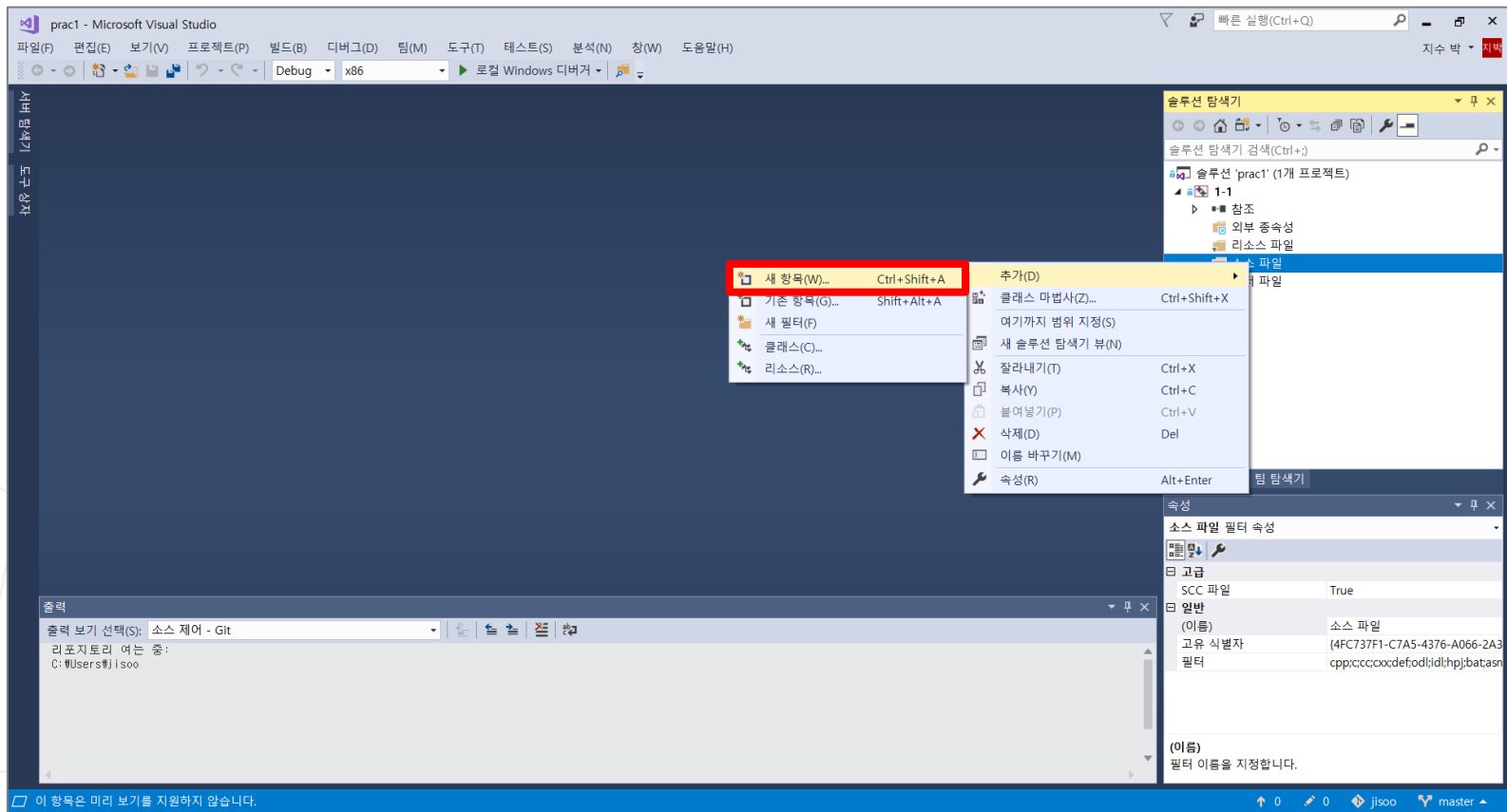
- Windows 데스크톱 프로젝트 → 빈 프로젝트(E) 체크 → 확인



# Visual Studio 2019 사용법 [7/13]

## ● Visual Studio 2019 항목 추가

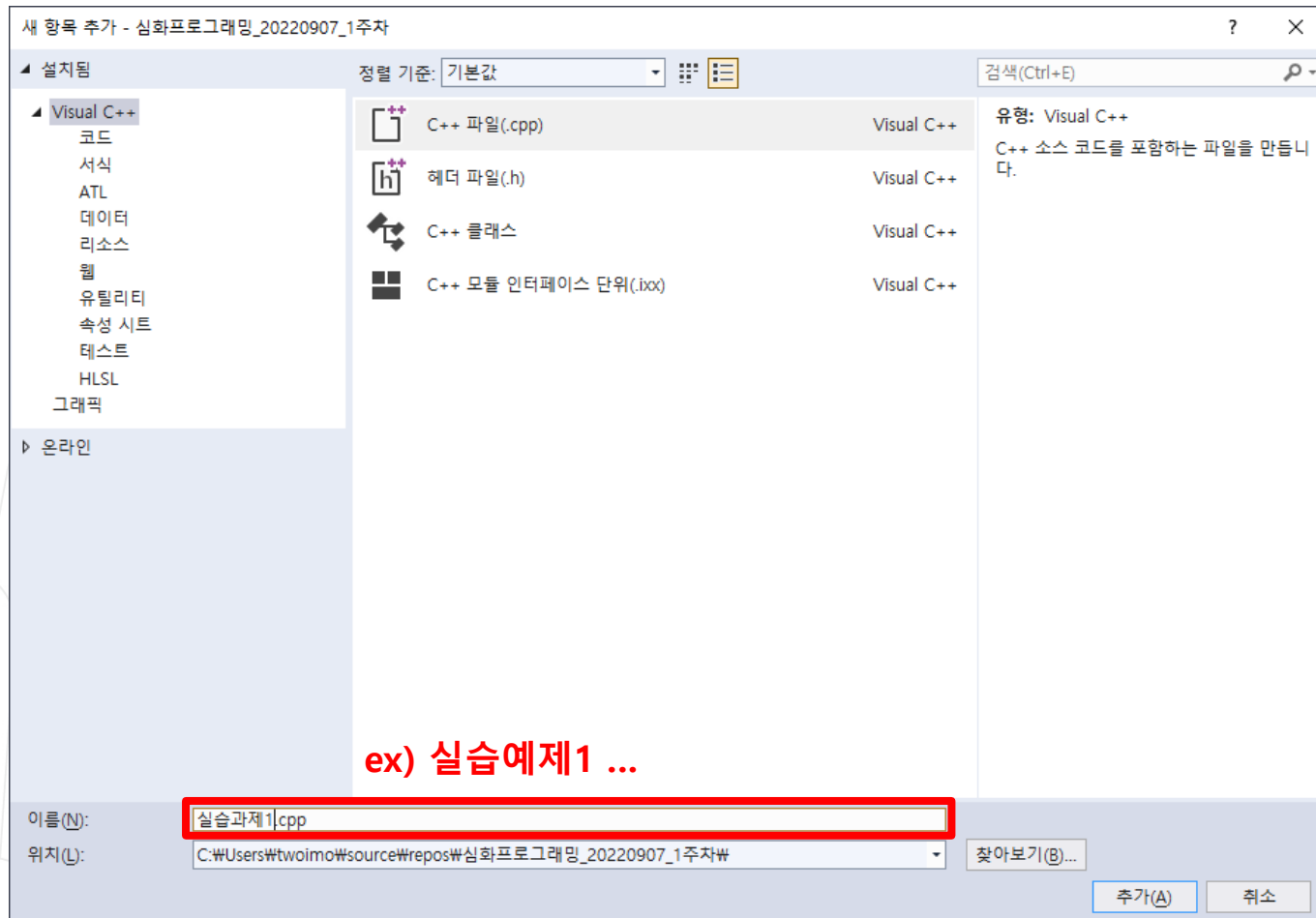
- 솔루션 탐색기 → 프로젝트 우클릭 → 추가(D) → 새 항목(W)
- 소스 파일 클릭 후 단축키(컨트롤 + 시프트 + A) 사용 가능



# Visual Studio 2019 사용법 [8/13]

## ● Visual Studio 2019 항목 추가

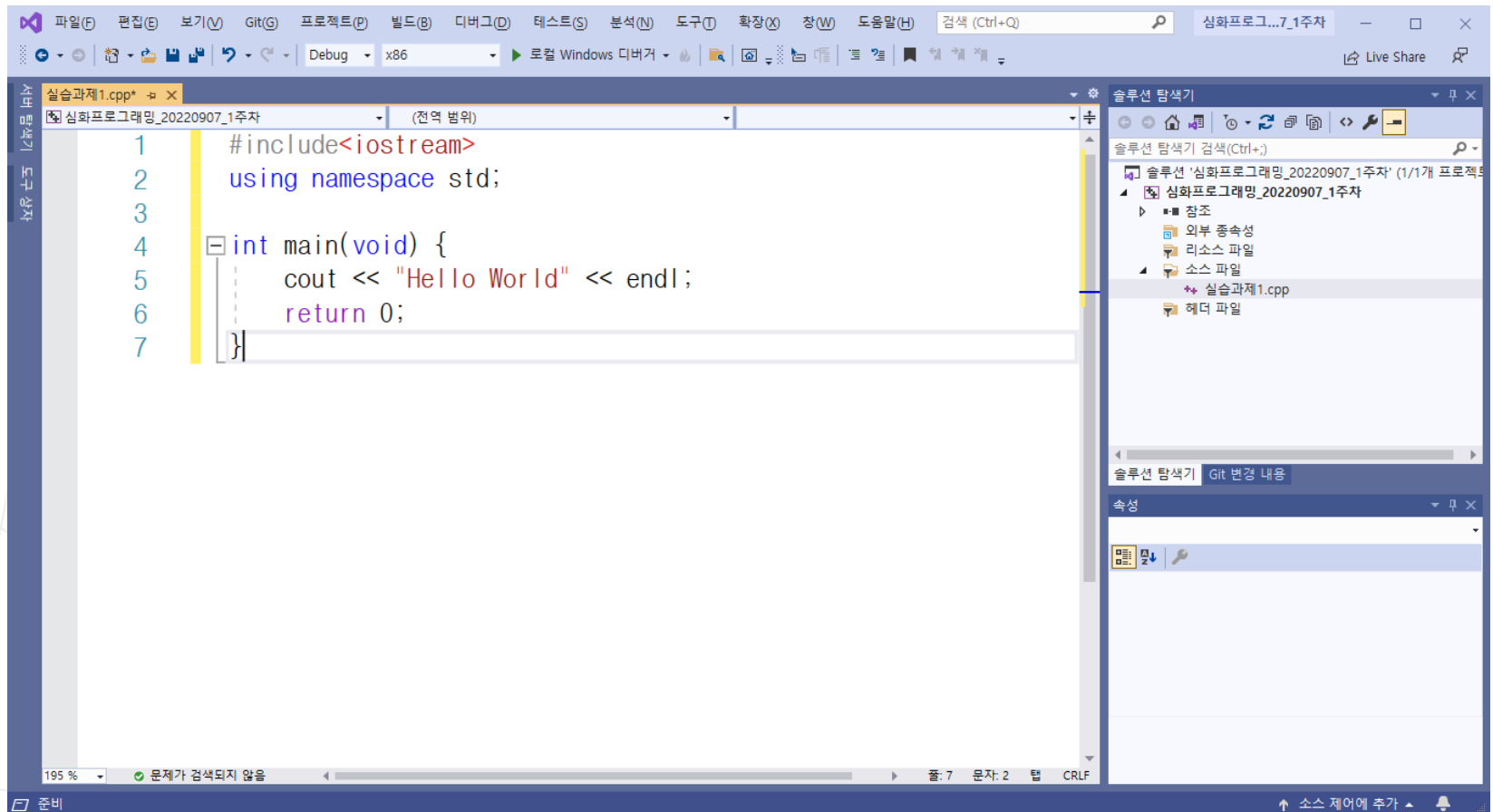
- Visual C++ → C++파일(.cpp) → 이름(N) 작성 → 추가(A)



# Visual Studio 2019 사용법 [9/13]

## ● Visual Studio 2019 항목 추가

- 컨트롤 + 마우스 스크롤을 통해 소스코드 글자크기 변경 가능. 예제 따라 해보기

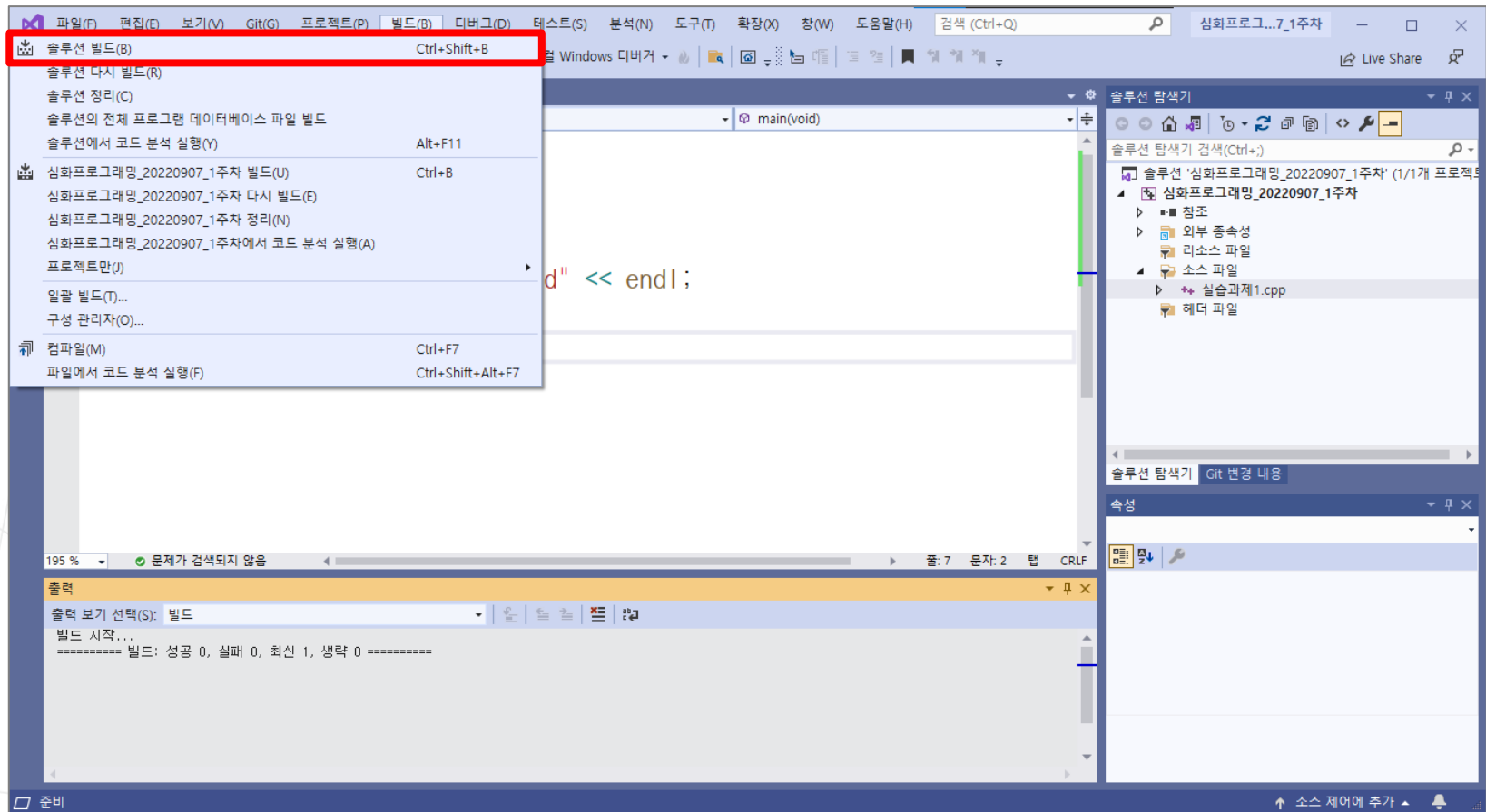




# Visual Studio 2019 사용법 [10/13]

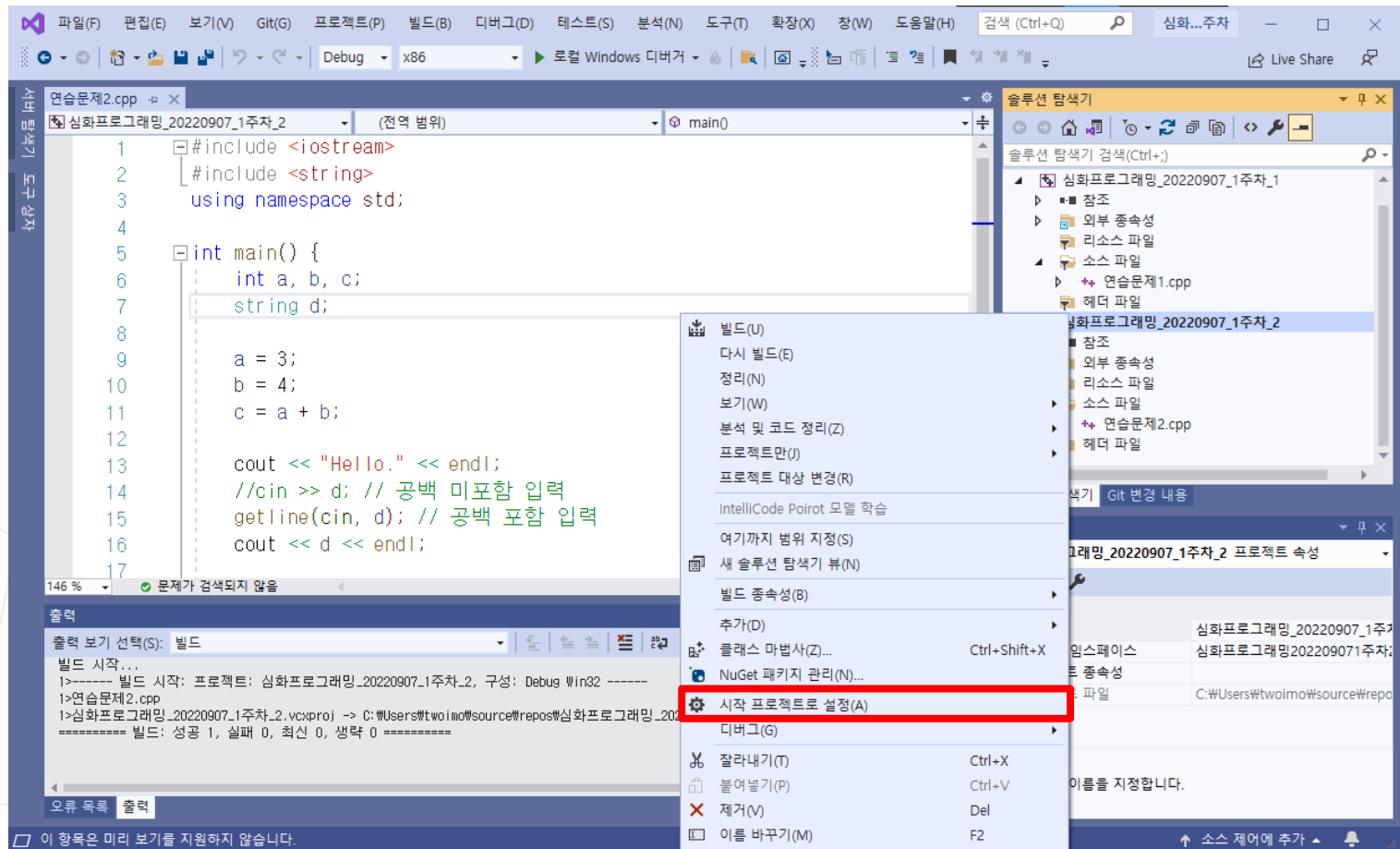
## ● Visual Studio 2019 항목 추가

- 빌드 단축키(컨트롤 + 시프트 + B)를 통해 소스코드 빌드 가능



# Visual Studio 2019 사용법 [11/13]

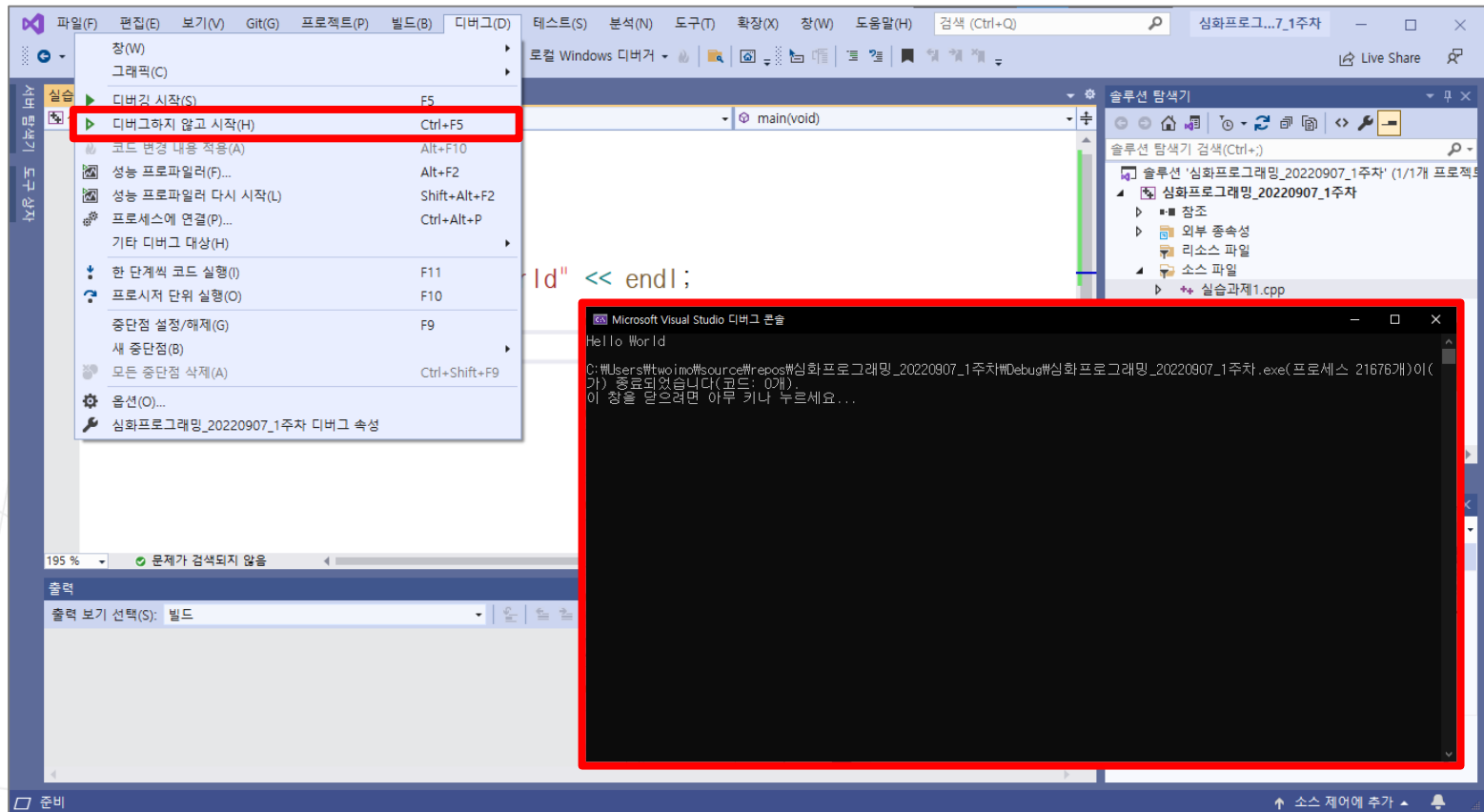
- Visual Studio 2019 다른 프로젝트 내 소스코드 실행 방법
  - 원하는 프로젝트 우클릭 → 시작 프로젝트로 설정(A) → 빌드 or 디버깅



# Visual Studio 2019 사용법 [13/13]

## ● Visual Studio 2019 프로그램 실행

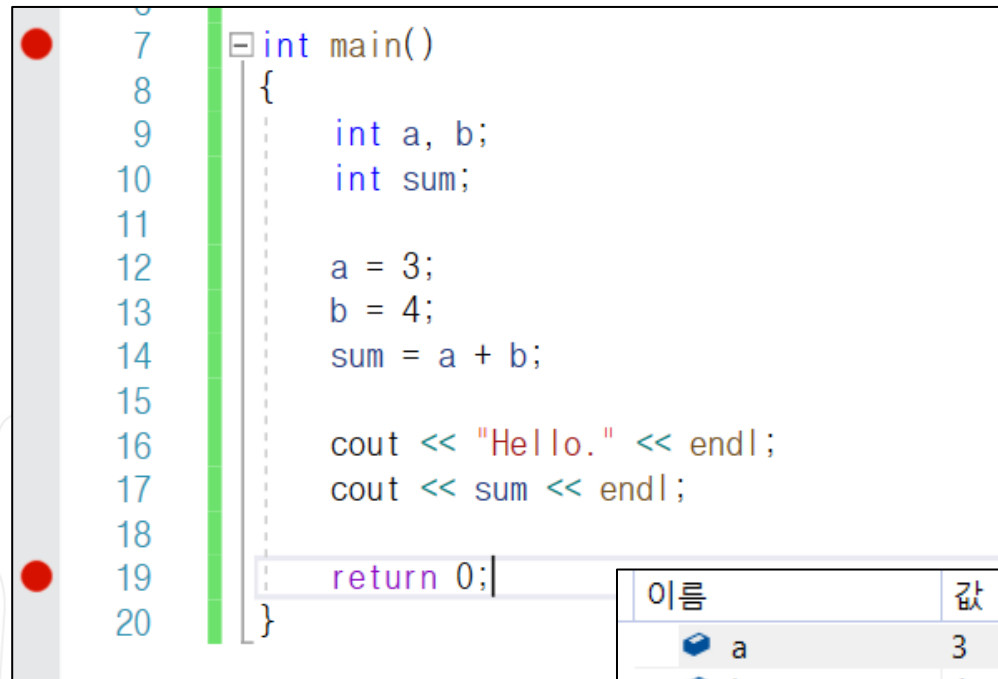
- 디버그(D) → 디버그하지 않고 시작(H) → 프로그램 실행
- 컨트롤 + F5 단축키를 통해서도 프로그램 실행 가능



# Visual Studio 2019 사용법 [12/13]




## ● Visual Studio 2019 디버깅

- 중단점 설정(단축키:F9) → 디버깅 시작(단축키:F5) → 한 단계씩(단축키: F11)



```
7  int main()  
8  {  
9      int a, b;  
10     int sum;  
11  
12     a = 3;  
13     b = 4;  
14     sum = a + b;  
15  
16     cout << "Hello." << endl;  
17     cout << sum << endl;  
18  
19     return 0;  
20 }
```

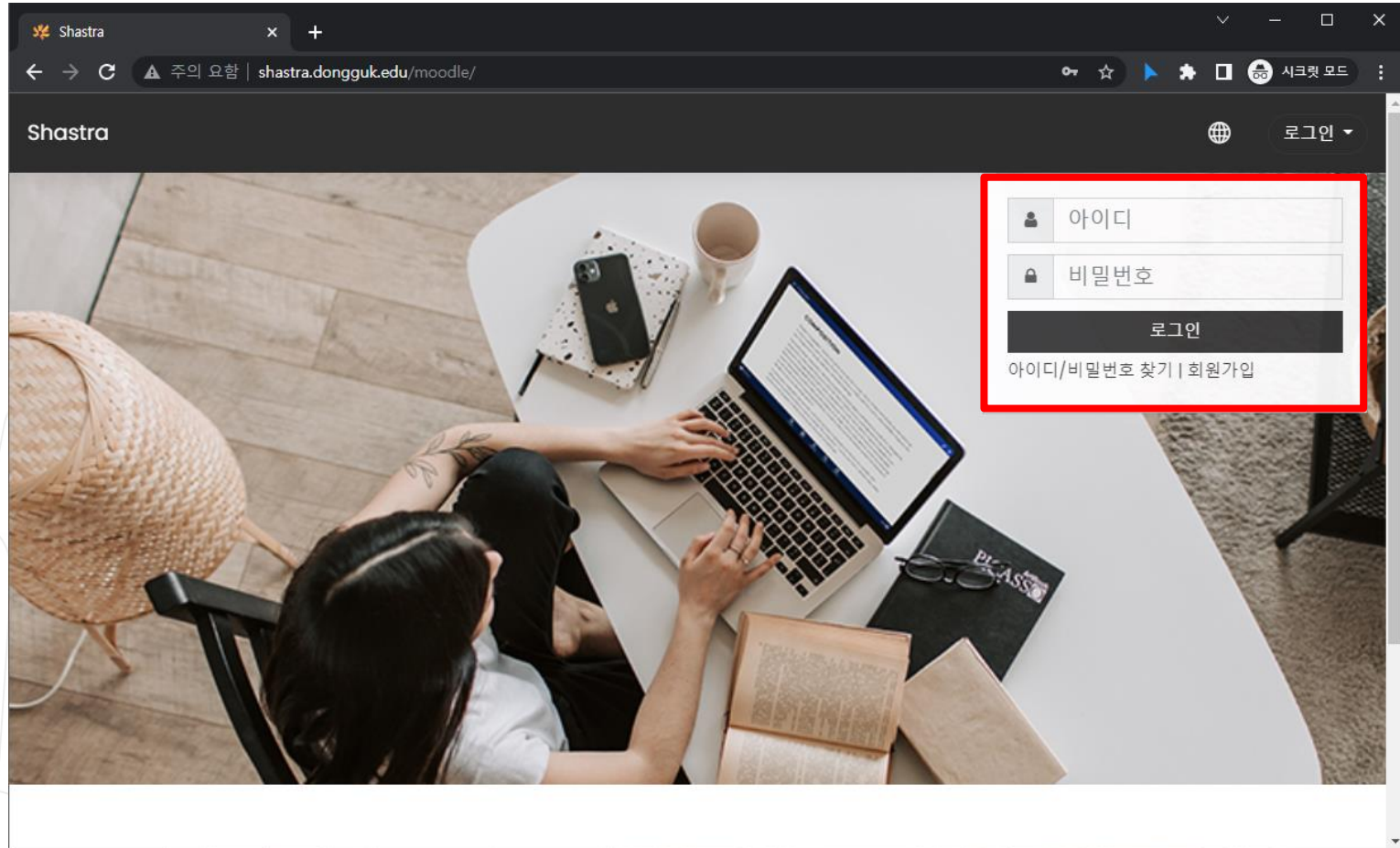
A red dot indicates a breakpoint is set on line 19. A green vertical bar highlights the current execution position at line 19.

이름	값	형식
 a	3	int
 b	4	int
 sum	7	int

# Sharstra 사용법 [1/5]

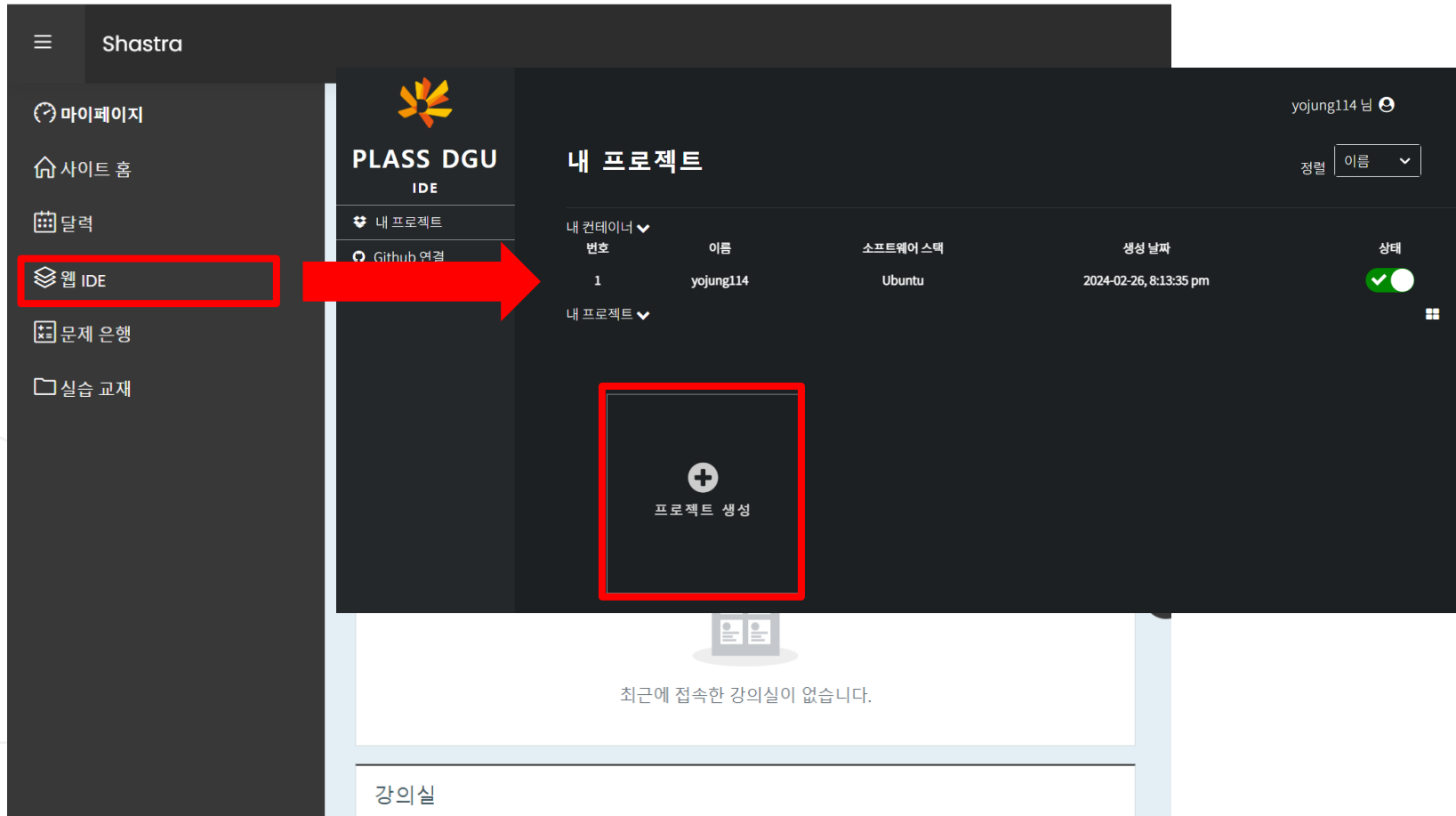
- 동국대학교 샤스트라 접속하기

- 홈페이지(<http://shastra.dongguk.edu/moodle/>) 접속 후 회원가입 및 로그인 진행



- 동국대학교 샤스트라 프로젝트 생성

- 샤스트라 로그인 → 웹 IDE 클릭 → 동국대학교 IDE 접속 → 프로젝트 생성



- 동국대학교 샤스트라 프로젝트 생성

- 프로젝트 정보 입력 → C++ 언어 선택 → 분야 선택 심화 프로그래밍 → 프로젝트 생성

The screenshot displays the PLASS DGU IDE interface. On the left, there's a sidebar with the PLASS DGU logo and navigation options like '내 프로젝트' (My Projects) and 'Github 연결' (Connect to Github). The main area is titled '내 프로젝트' (My Projects) and shows a list of projects. A modal dialog box titled '프로젝트 생성' (Create Project) is open in the center. It prompts the user to '프로젝트 정보를 입력해주세요.' (Please enter project information). The form includes a text input for the project name (filled with '심화프로그래밍\_실습과제\_1주차'), dropdown menus for '언어 선택' (Language Selection) set to 'C++' and '분야 선택' (Field Selection) set to '심화 프로그래밍', and another dropdown for '용도 선택' (Purpose Selection) set to '자료구조 학습용' (For learning data structures). There's also a '태그명 입력' (Enter tag name) section with a placeholder '태그를 입력하고 Enter 혹은 추가 버튼을 눌러주세요.(1개씩 3)' (Enter tags and press Enter or the Add button. (1 tag at a time, 3)). A '프로젝트 생성' (Create Project) button is at the bottom of the dialog. On the right side of the IDE, there's a user profile 'yojung114 님' and a '정렬' (Sort) dropdown set to '이름' (Name). A '상태' (Status) indicator shows a green checkmark.

# Sharstra 사용법 [4/5]

- 동국대학교 샤스트라 프로젝트 실행
  - 생성한 프로젝트 클릭 실행(Ubuntu, VSCode, Gcc ...)

The screenshot shows the PLASS DGU IDE interface. On the left is a sidebar with the PLASS DGU logo and navigation links for '내 프로젝트' and 'Github 연결'. The main area is titled '내 프로젝트' and displays a table of projects. The table has columns for '번호', '이름', '소프트웨어 스택', '생성 날짜', and '상태'. A single project is listed with the number '1', name 'yojung114', stack 'Ubuntu', creation date '2024-02-26, 8:13:35 pm', and a status of 'ON'. Below the table is a section for '내 프로젝트' with two options: '프로젝트 생성' (indicated by a plus icon) and a highlighted project card. The highlighted project card features a C++ icon and the text '심화프로그래밍\_실습과제\_1주차' and '날짜 : 2024-03-04'.

번호	이름	소프트웨어 스택	생성 날짜	상태
1	yojung114	Ubuntu	2024-02-26, 8:13:35 pm	ON

내 프로젝트 ▼

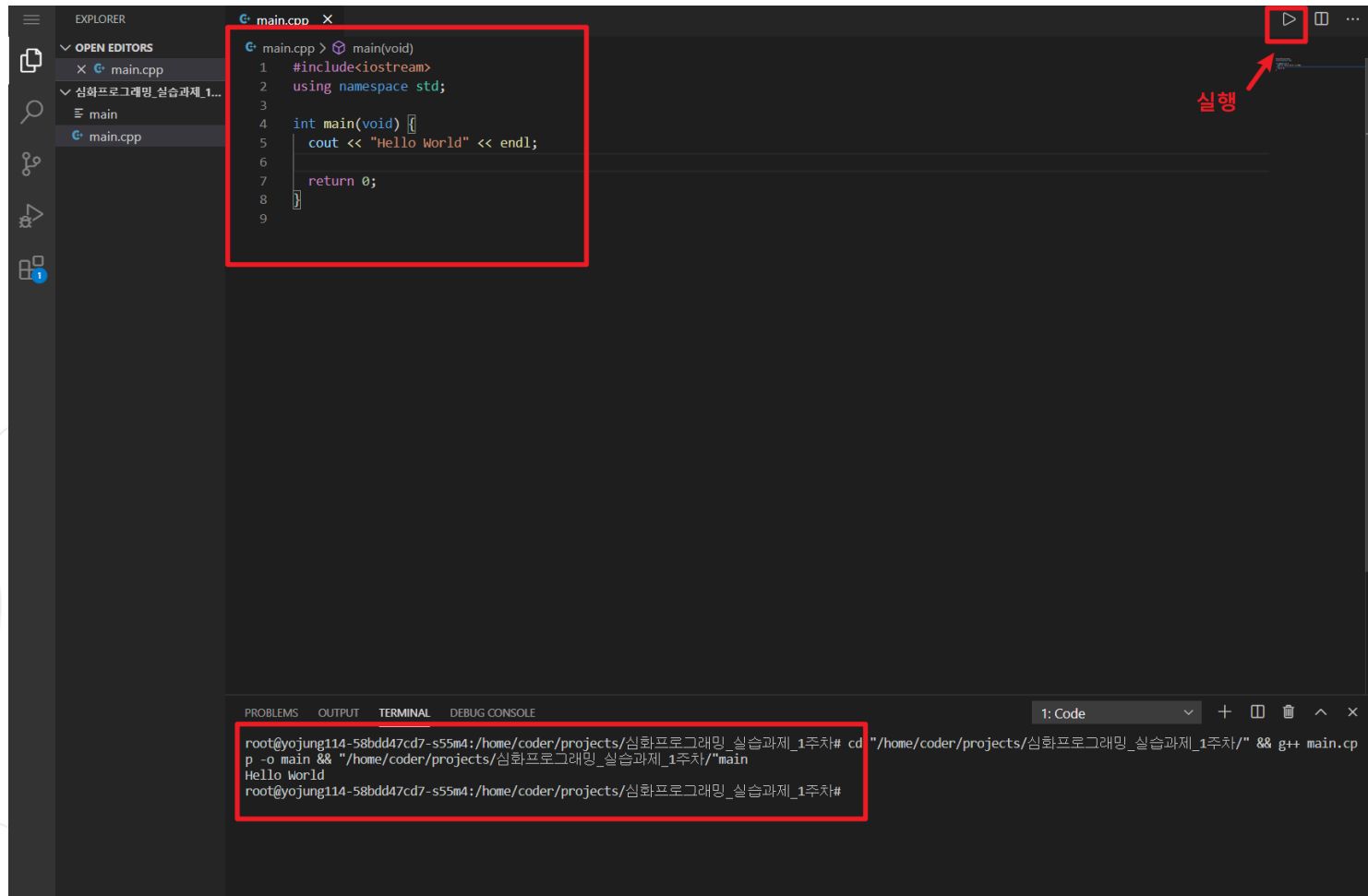
프로젝트 생성

심화프로그래밍\_실습과제\_1주차  
날짜 : 2024-03-04



# Sharstra 사용법 [5/5]

- 동국대학교 샤스트라 프로젝트 실행
  - 웹 브라우저를 통해 실습강의를 따라올 수 있으나 컴파일러(gcc) 유의



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the file structure. The main editor displays a C++ program in `main.cpp`. The program is enclosed in a red box. The code is as follows:

```
1 #include<iostream>
2 using namespace std;
3
4 int main(void) {
5     cout << "Hello World" << endl;
6
7     return 0;
8 }
9
```

The Run and Debug panel on the right shows the execution of the program. The output is displayed in the terminal, which is also enclosed in a red box. The output is:

```
root@yojung114-58bdd47cd7-s55m4:/home/coder/projects/심화프로그래밍_실습과제_1주차# cd "/home/coder/projects/심화프로그래밍_실습과제_1주차/" && g++ main.cpp -o main && "/home/coder/projects/심화프로그래밍_실습과제_1주차/"main
Hello World
root@yojung114-58bdd47cd7-s55m4:/home/coder/projects/심화프로그래밍_실습과제_1주차#
```

# 1주차 실습과제

- 사용자와 프로그램이 가위, 바위, 보 게임을 할 수 있는 프로그램을 작성하시오.
- 함수 자신이 호출된 횟수를 반환하는 함수를 작성하여 이를 검사하는 프로그램을 작성하시오.
- 1에서 100까지의 하나의 난수를 발생하여 얻어낸 값을 사용자가 맞추는 프로그램을 작성하시오. 단 사용자가 입력한 정수가 정해진 난수와 맞지 않은 경우, 입력할 값의 정보를 다음과 같이 출력하여, 정확한 값을 사용자가 맞출 때까지 계속 답을 요구하도록 하며 다음 조건을 충족하도록 한다.

① 함수 `setNumber()`에서는 1에서 100까지의 하나의 난수를 발생하여 전역변수 `number`에 저장한다.

② 시스템이 정한 `number`를 사용자가 맞출 때까지 계속 진행한다.

③ 사용자가 정답을 맞추지 못하는 경우는 힌트를 자세히 준다.

④ 힌트를 주기 위하여 변수 `min`, `max`를 이용하며, 이 변수는 정적 외부변수로 선언한다.

⑤ 함수는 `main()`과 함수 `setNumber()`, `printHead()`, `printHigher()`, `printLower()`, `printAnswer()`로 구성한다.

⑥ 사용자가 정답을 맞추기 위하여 시도한 횟수를 저장하는 변수 `trycount`를 이용하여, 매번 이 값이 출력되도록 한다.



# 1주차 실습과제

- 다음 배열에서 모든 원소의 값을 모두 n씩 증가시키는 함수를 작성하시오.

```
int data[ ] = {3, 21, 35, 57, 24, 82, 8};
```

- 함수의 형식 인자로 다음과 같이 기술된 함수 swap()은 호출하는 부분에서 실 인자인 두 정수를 서로 바꾸는 기능을 수행한다. 함수 swap()을 구현하여 결과를 알아보는 프로그램을 작성하시오.

```
void swap(int *x, int *y)
{ ... }
```

- 두 이차원 배열의 더하기와 빼기를 수행하는 함수를 만들어, 다음 배열의 연산 결과를 알아보시오.

4.2	4.3	3.8	+	(-)	5.2	2.1	1.8
3.7	1.5	0.7			3.2	1.4	2.9

# 1주차 실습과제

- 배열 포인터를 이용하여 다음의 이차원 배열에서 모든 원소의 값을 5씩 증가시키는 프로그램을 작성하시오.

```
int ary[ ][5] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
배열 포인터 : int (*ptr)[5];
```

- 자료유형 double형 1차원 배열을 다음과 같이 초기화하고, 첫 번째 인자인 배열 source를 두 번째 인자인 배열 target에 복사하는 함수를 만들어 결과를 알아보는 프로그램을 작성하시오.

```
double ary[5] = {3.12, 5.14, 7.25, 7.48, 5.91} ;
```

```
copyarray(double *source, double *target, int size) ;
```

- 다음과 같이 int 형 변수 i에 0x324F3A24을 정의하여 char \* 변수 pc를 선언하여 pc 변수를 이용하여 i의 저장 값을 16 진수 324F3A24 로 출력하는 프로그램을 작성하시오.

```
int n = 0x324F3A24 ;
```