

# Visual Studio Session

---

Youngsup Kim  
idebtor@gmail.com

# Contents

---

I. How to Install Visual Studio Professional 2019

II. Let's build your first c++ project and debug it

III. How to handle external library in Visual Studio

# Before you start

Clone this Repository!

<https://github.com/SukJinKim/19-1-Visual-Studio-Session>

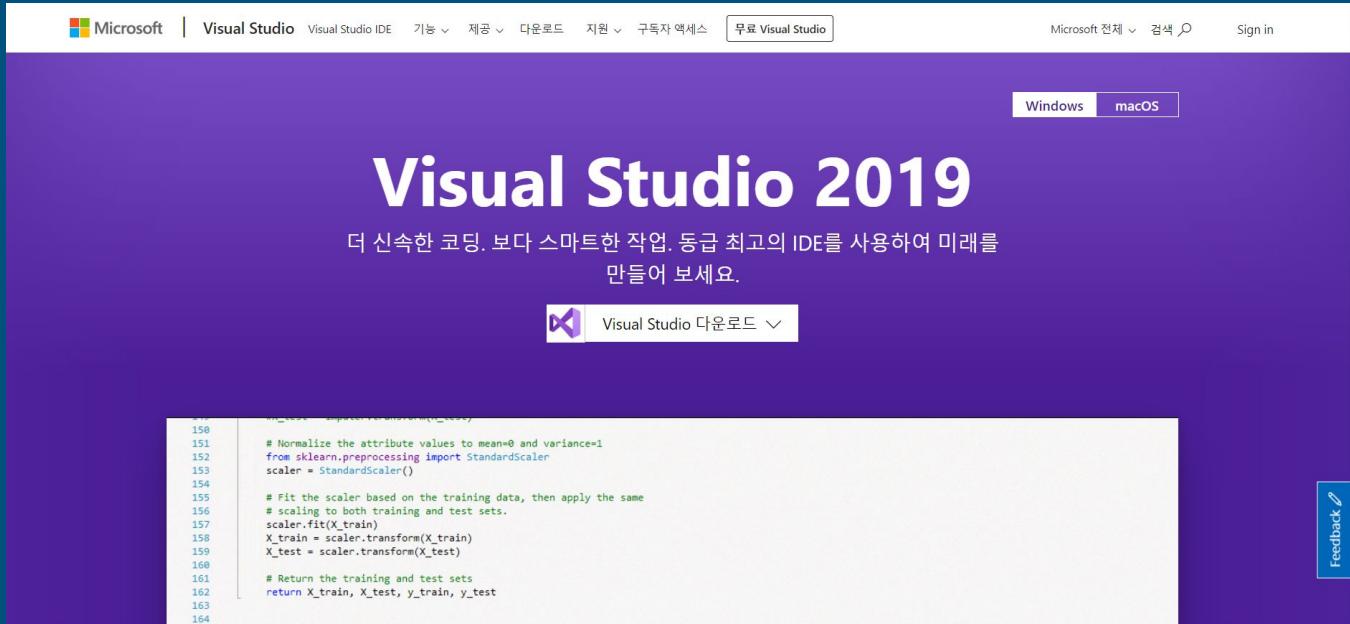


# How to install Visual Studio Professional 2019

# How to install Visual Studio Professional 2019

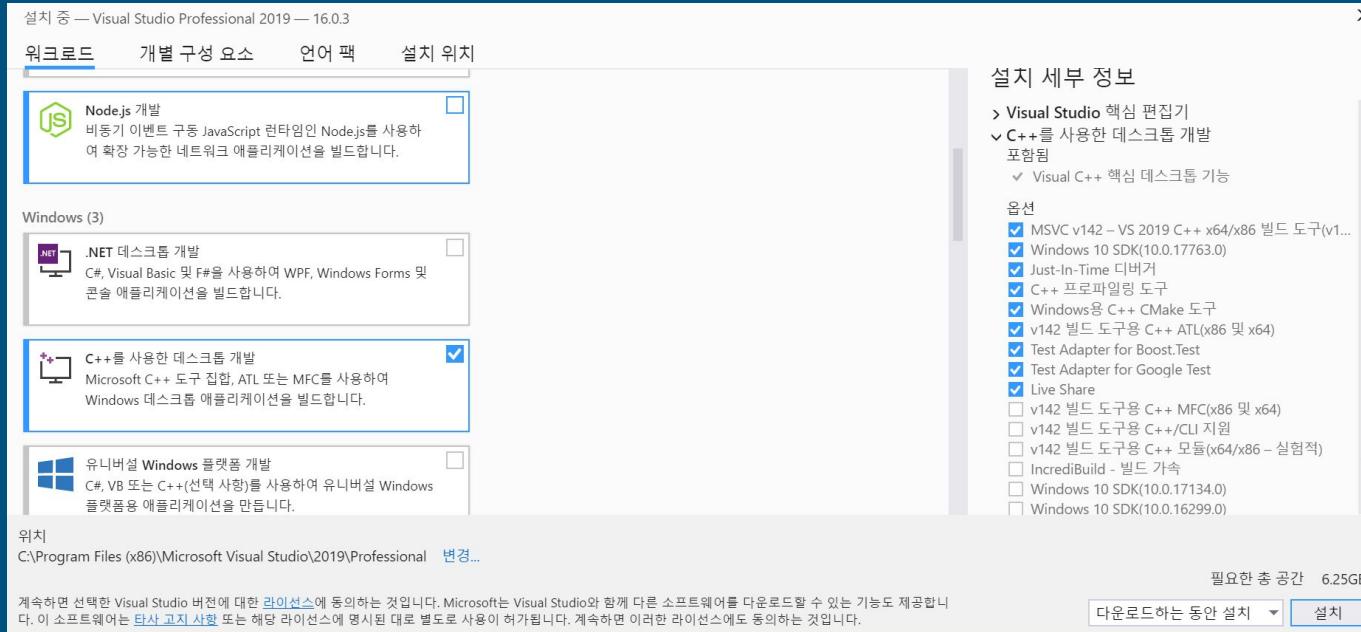
---

## 1. Download Visual Studio Professional 2019



# How to install Visual Studio Professional 2019

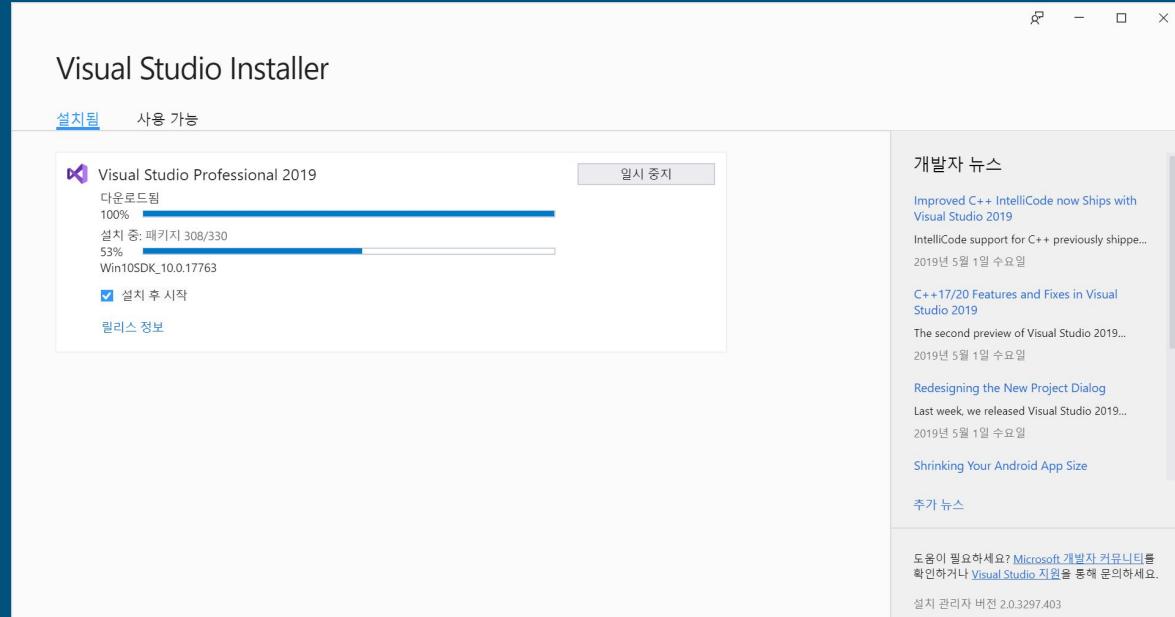
## 2. Check the box labeled desktop development using c++.



# How to install Visual Studio Professional 2019

---

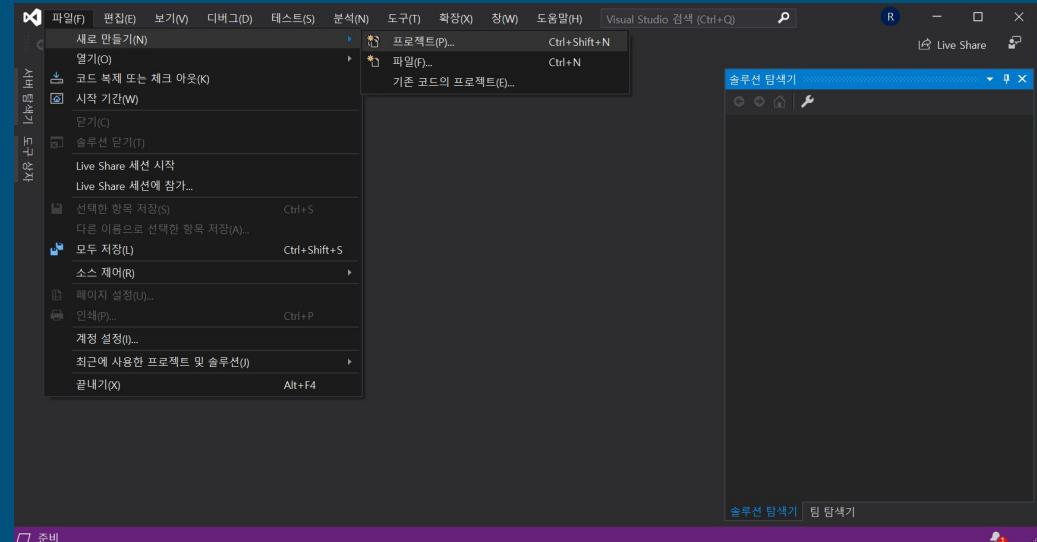
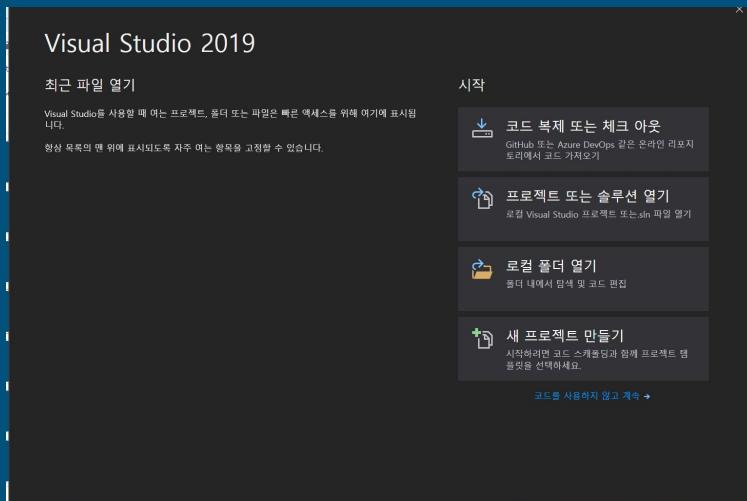
## 3. Wait until the installation is completed.



Let's build your first c++ project  
and debug it!

# Your first C++ console project - Create a project

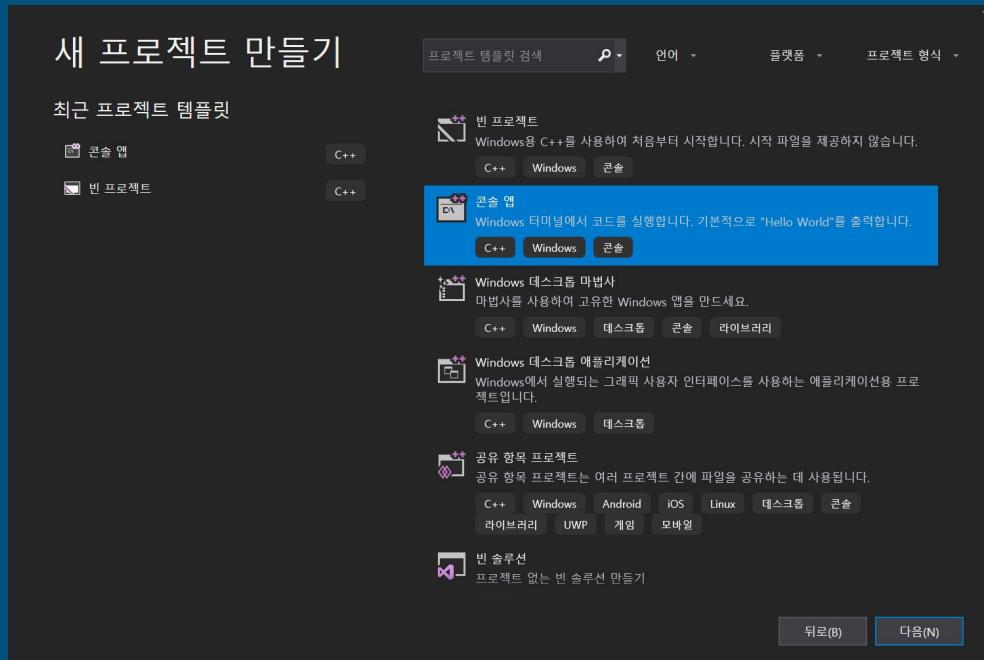
1. Choose **Create a new project** to get started. Otherwise, on the menubar in Visual Studio, choose **File > New > Project**. The Create a new project window opens.



# Your first C++ console project - Create a project

---

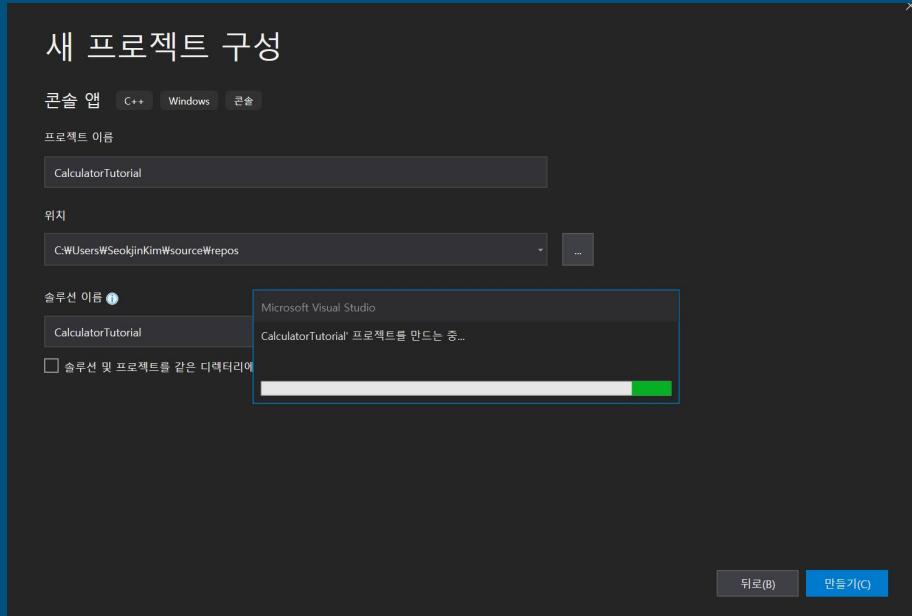
2. In the list of project templates, choose **Console App**, then choose Next



# Your first C++ console project - Create a project

---

3. In the Configure your new project dialog box, select the Project name edit box, name your new project **CalculatorTutorial**, then choose Create.



# Your first C++ console project - Create a project

The screenshot shows the Microsoft Visual Studio interface with a dark theme. The main window displays the code for 'CalculatorTutorial.cpp'.

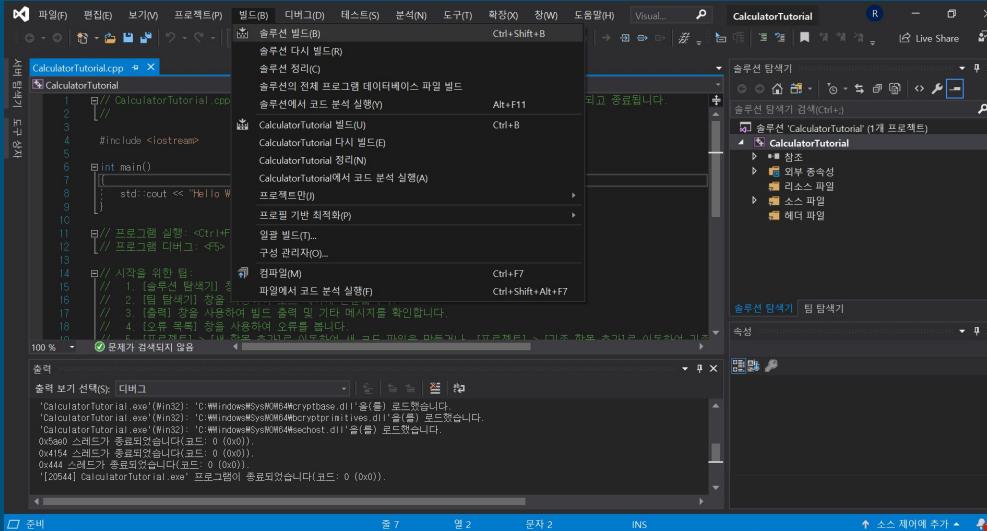
```
1 // CalculatorTutorial.cpp : 이 파일에는 main 함수가 포함됩니다. 여기서 프로그램 실행이 시작되고 종료됩니다.
2
3 //include <iostream>
4
5 int main()
6 {
7     std::cout << "Hello World!\n";
8 }
9
10 // 프로그램 실행: <Ctrl+F5> 또는 [디버그] > [디버깅하지 않고 시작] 메뉴
11 // 프로그램 디버그: <F5> 키 또는 [디버그] > [디버깅 시작] 메뉴
12
13 // 시작을 위한 팀:
14 // 1. [솔루션 탐색기] 창을 사용하여 파일을 추가/관리합니다.
15 // 2. [팀 탐색기] 창을 사용하여 소스 제어에 연결합니다.
16 // 3. [솔직] 창을 사용하여 빌드 출력 및 기타 메시지를 확인합니다.
17 // 4. [오류 목록] 창을 사용하여 오류를 봅니다.
18
19 // 코드 편집기 창은 기본으로 사용되어 코드 파일을 만들거나 편집하는 데 도움이 됩니다.
20
```

The status bar at the bottom indicates: 'CalculatorTutorial' 프로젝트를 만들고 있습니다... 프로젝트를 만들었습니다.' (Creating the CalculatorTutorial project... Project created.)

# Your first C++ console project - build and run

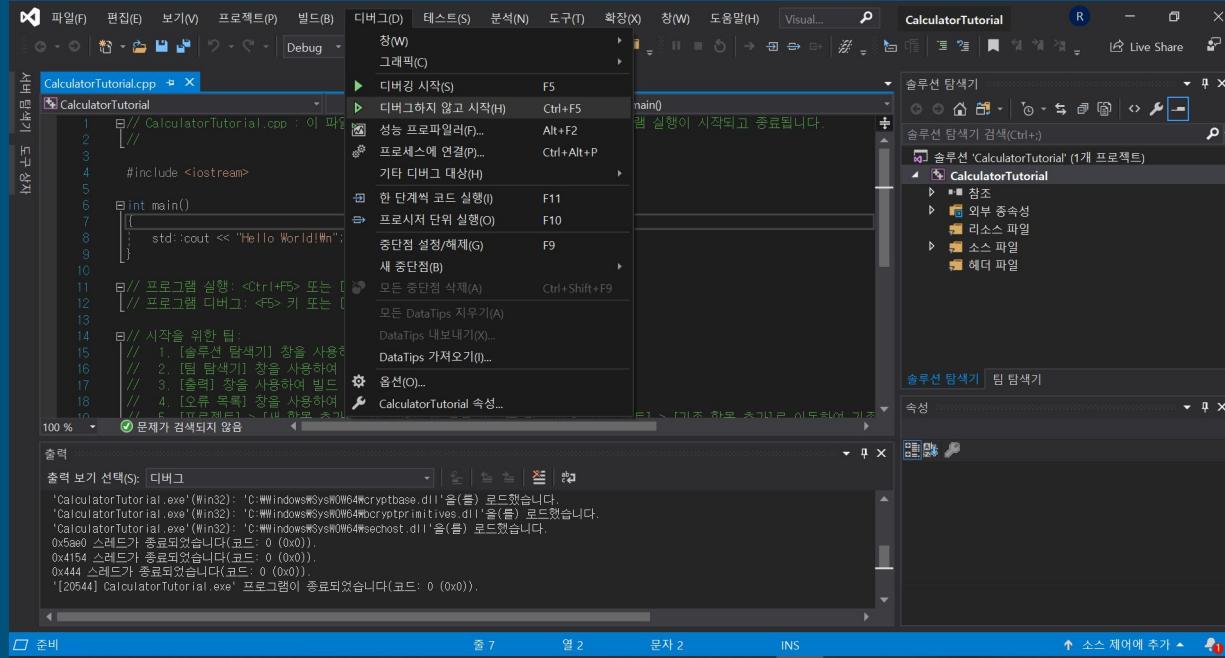
Let's verify that your new app builds and run!

1. To build your project, choose Build Solution from the Build menu. The Output window shows the results of the build process.



# Your first C++ console project - build and run

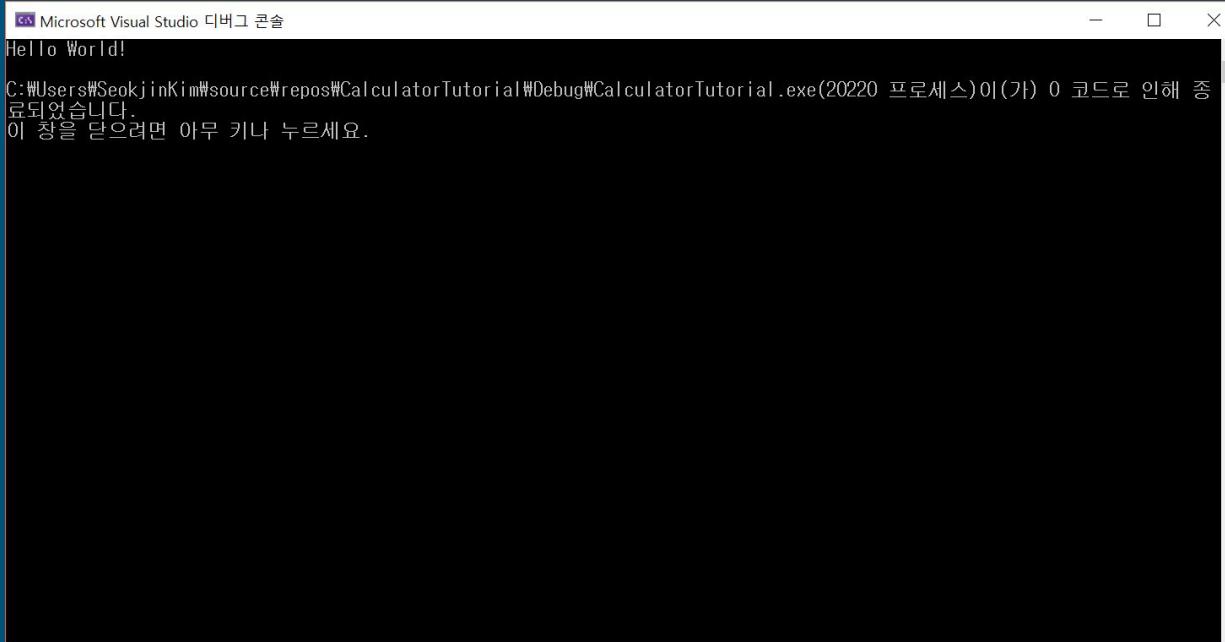
2. To run the application, press Ctrl+F5 or go to the Debug menu and choose Start Without Debugging.



# Your first C++ console project - build and run

---

2. To run the application, press Ctrl+F5 or go to the Debug menu and choose Start Without Debugging.



# Your first C++ console project - Edit the Code

1. Edit the code to match *CalculatorTutorial1.cpp* in github.

The screenshot shows the Visual Studio IDE interface with the following details:

- MenuBar:** 파일(F), 편집(E), 보기(V), 프로젝트(P), 빌드(B), 디버그(D), 테스트(S), 분석(N), 도구(T), 확장(X), 창(W), 도움말(H), Visual...
- Toolbars:** Debug, x86, 로컬 Windows 디버거, Live Share
- Code Editor:** 파일명: CalculatorTutorial.cpp, 내용은 main() 함수와 사용자 입력을 처리하는 코드입니다.
- Solution Explorer:** 솔루션 탐색기, 솔루션 'CalculatorTutorial' (1개 프로젝트)
  - CalculatorTutorial
    - 참조
    - 외부 종속성
    - 리소스 파일
    - 소스 파일
      - CalculatorTutorial.cpp
      - 헤더 파일
- Output Window:** 출력, 빌드 시작: 프로젝트: CalculatorTutorial, 구성: Debug Win32  
1>calculatorTutorial.cpp  
1>calculatorTutorial.vcxproj -> C:\Users\SeokjinKim\source\repos\CalculatorTutorial\Debug\CalculatorTutorial.exe  
===== 빌드: 성공 1, 실패 0, 최신 0, 생략 0 =====
- Status Bar:** 준비

# Your first C++ console project - Add code to do some math.

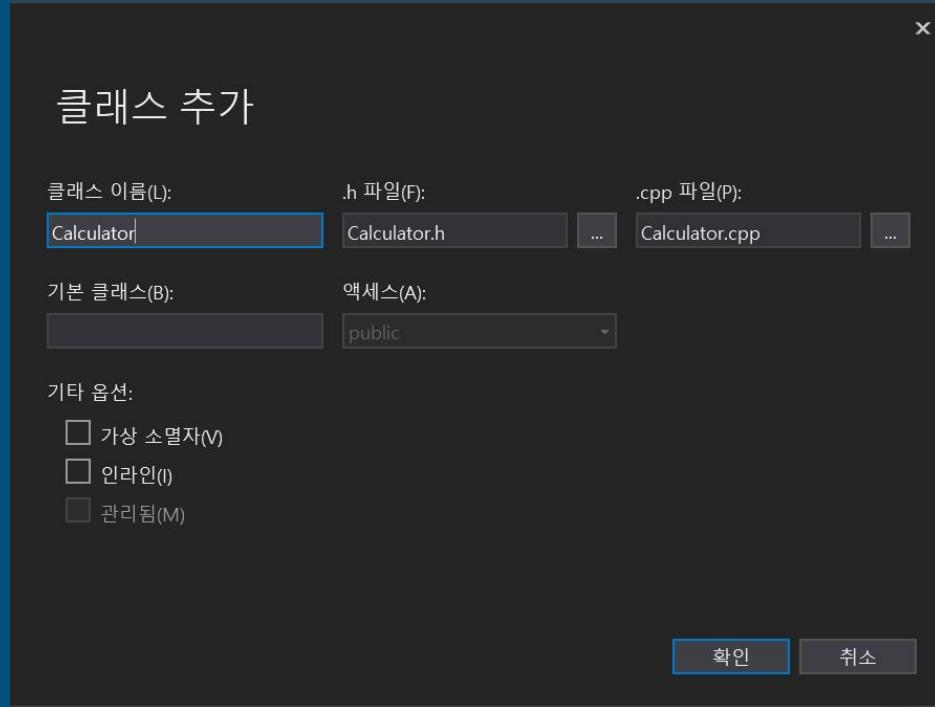
---

1. Go to the Project menu and choose Add Class. In the Class Name edit box, enter *Calculator*. Choose OK.

Two new files get added to your project. To save all your changed files at once, press Ctrl+Shift+S. It's a keyboard shortcut for File > Save All. There's also a toolbar button for Save All, an icon of two floppy disks, found beside the Save button. In general, it's good practice to do Save All frequently, so you don't miss any files when you save.

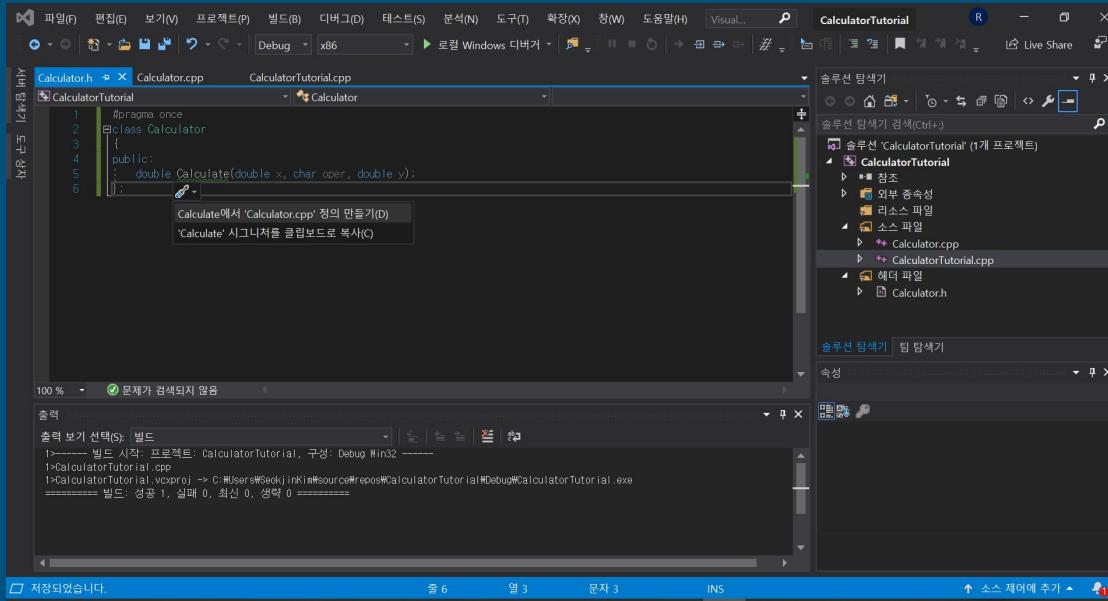
# Your first C++ console project - Add code to do some math.

---



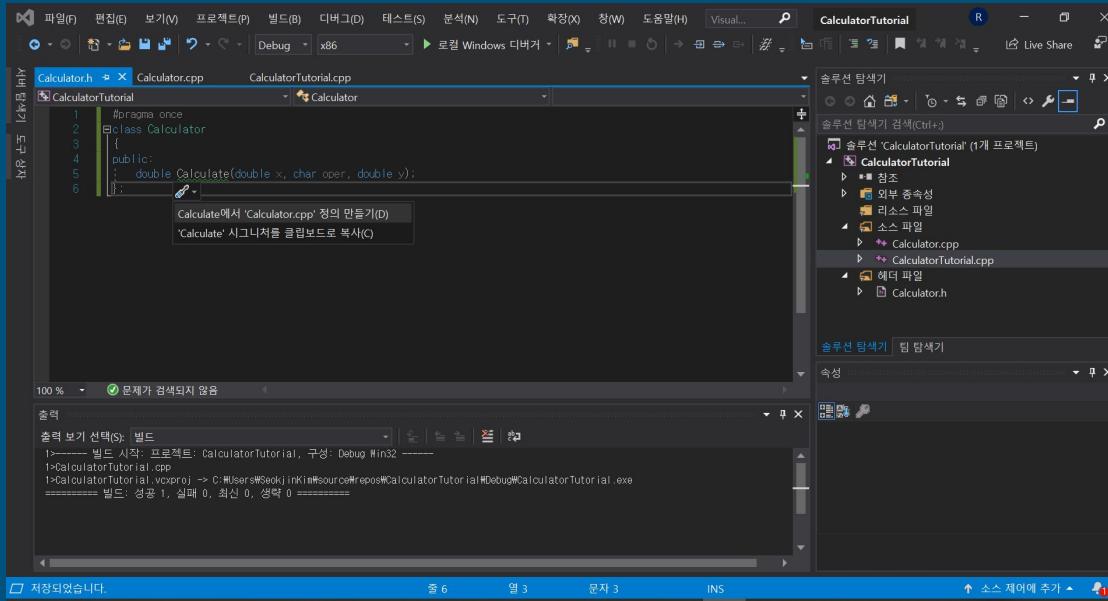
# Your first C++ console project - Add code to do some math.

3. You'll see a green squiggle appear under Calculate. It's because we haven't defined the Calculate function in the .cpp file. Hover over the word, click the lightbulb (in this case, a screwdriver) that pops up, and choose Create definition of 'Calculate' in Calculator.cpp.



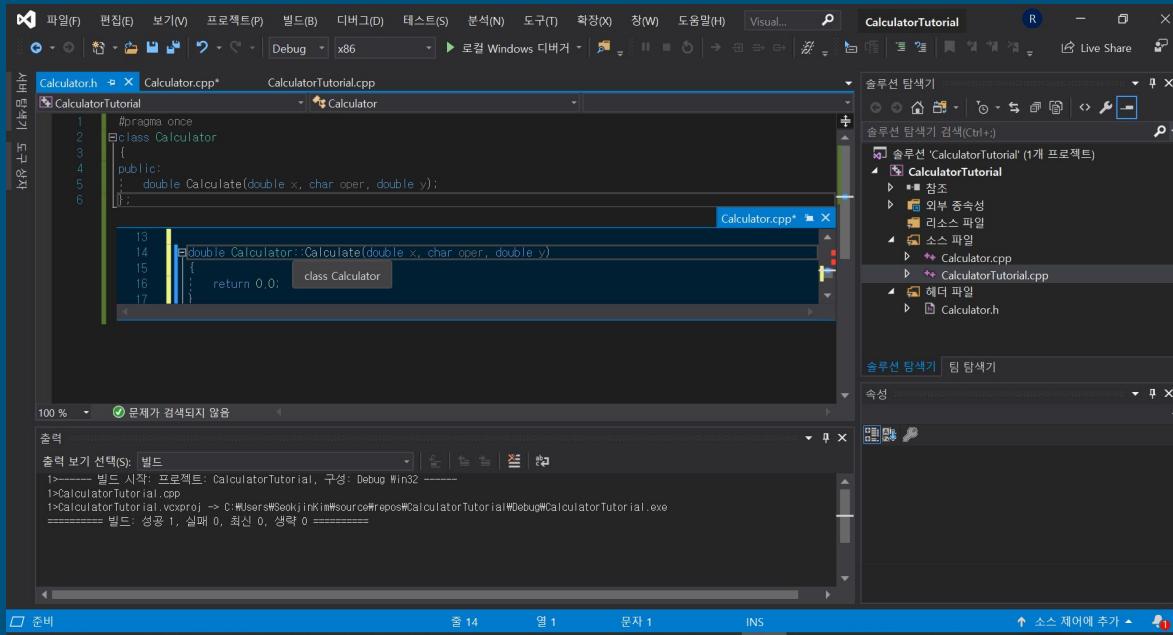
# Your first C++ console project - Add code to do some math.

3. You'll see a green squiggle appear under Calculate. It's because we haven't defined the Calculate function in the .cpp file. Hover over the word, click the lightbulb (in this case, a screwdriver) that pops up, and choose Create definition of 'Calculate' in Calculator.cpp.



# Your first C++ console project - Add code to do some math.

4. A pop-up appears that gives you a peek of the code change that was made in the other file. The code was added to *Calculator.cpp*. Currently, it just returns 0.0. Let's change that. Press Esc to close the pop-up.



# Your first C++ console project - Add code to do some math.

5. Switch to the **Calculator.cpp** file in the editor window. Remove the Calculator() and ~Calculator() sections (as you did in the .h file) and edit the code to match **Calculator.cpp** in github.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure for "CalculatorTutorial". It includes a folder named "CalculatorTutorial" containing "Calculator.h", "Calculator.cpp", and "CalculatorTutorial.cpp". There is also a "header" folder containing "Calculator.h".
- Editor:** The active file is "Calculator.cpp". The code is as follows:

```
#include "Calculator.h"

double Calculator::Calculate(double x, char oper, double y)
{
    switch (oper)
    {
        case '+':
            return x + y;
        case '-':
            return x - y;
        case '*':
            return x * y;
        case '/':
            return x / y;
        default:
            return 0.0;
    }
}
```

The status bar at the bottom shows the build output:

```
빌드 시작: 프로젝트: calculatorTutorial, 구성: Debug Win32
1>calculatorTutorial.cpp
1>calculatorTutorial.vcxproj -> C:\Users\Seokjin\minisource\repos\CalculatorTutorial\Debug\CalculatorTutorial.exe
===== 빌드: 성공 1, 실패 0, 최신 0, 생략 0 =====
```

# Your first C++ console project - Call the Calculator class member functions

- Now let's update the main function in *CalculatorTutorial.cpp* to match *CalculatorTutorial2.cpp* in github.

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Code Editor:** The main window displays *CalculatorTutorial.cpp* with the following content:

```
1 // CalculatorTutorial.cpp : This file contains the 'main' function. Program execution begins and ends there.
2
3
4 #include <iostream>
5 #include "Calculator.h"
6
7 using namespace std;
8
9 int main()
10 {
11     double x = 0.0;
12     double y = 0.0;
13     double result = 0.0;
14     char oper = '+';
15
16     cout << "Calculator Console Application" << endl << endl;
17     cout << "Please enter the operation to perform. Format: a+b | a-b | a*b | a/b"
18     << endl;
```

- Solution Explorer:** Shows the project structure for *CalculatorTutorial* with files *Calculator.cpp*, *CalculatorTutorial.cpp*, and *Calculator.h*.
- Task List:** Displays build logs for the project.

# Your first C++ console project - Call the Calculator class member functions

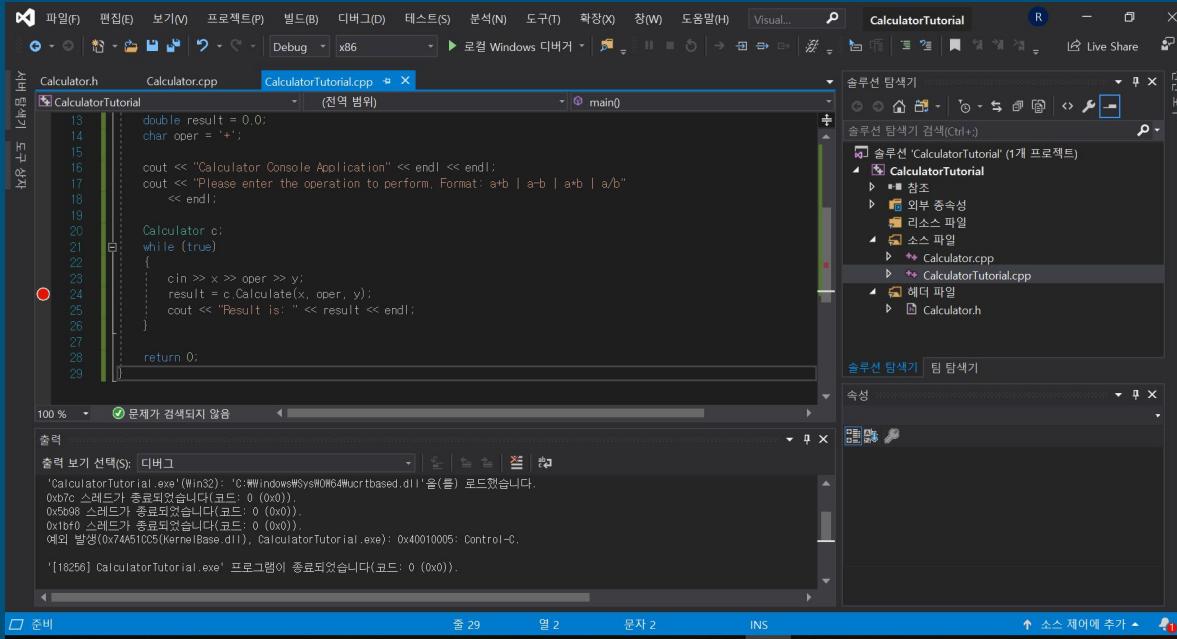
---

2. Build and test the code! Check the result when entered **12/0!** It's definitely a bug!!!

```
C:\Users\SeokjinKim\source\repos\CalculatorTutorial\Debug\CalculatorTutorial.exe
Calculator Console Application
Please enter the operation to perform. Format: a+b | a-b | a*b | a/b
5+5
Result is: 10
10-2
Result is: 8
11*3
Result is: 33
12/0
Result is: inf
```

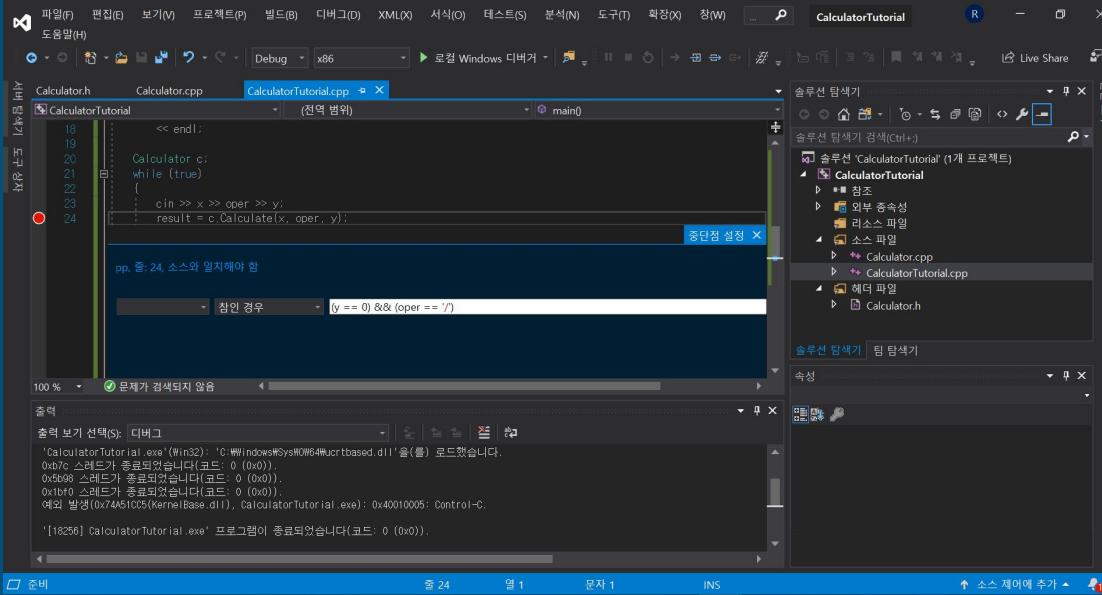
# Your first C++ console project - Debug the app

1. Set a breakpoint on the **result = c.Calculate(x, oper, y);** line, just after the user was asked for input. To set the breakpoint, click next to the line in the gray vertical bar along the left edge of the editor window. A red dot appears.



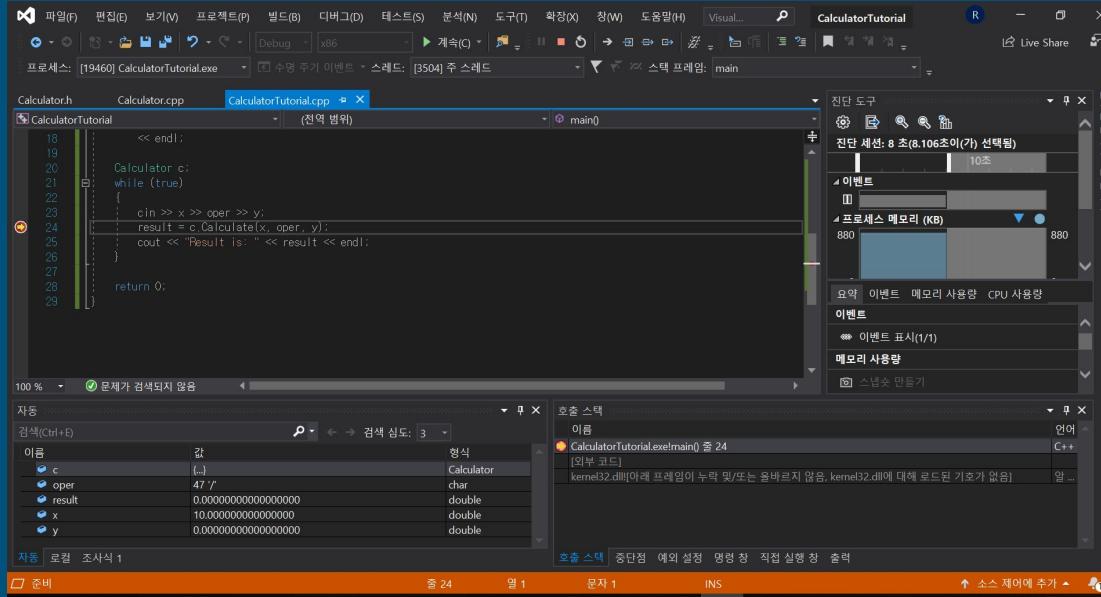
# Your first C++ console project - Debug the app

2. Right-click the red dot that represents the breakpoint, and choose **Conditions**. In the edit box for the condition, enter **(y == 0) && (oper == '/')**. Choose the Close button when you're done. The condition is saved automatically. Now we pause execution at the breakpoint specifically if a division by 0 is attempted.



# Your first C++ console project - Debug the app

3. To debug the program, press F5, or choose the Local Windows Debug toolbar button that has the green arrow icon. In your console app, if you enter something like "5 - 0", the program behaves normally and keeps running. However, if you type "10 / 0", it pauses at the breakpoint.



# Your first C++ console project - Useful windows in the debugger

---

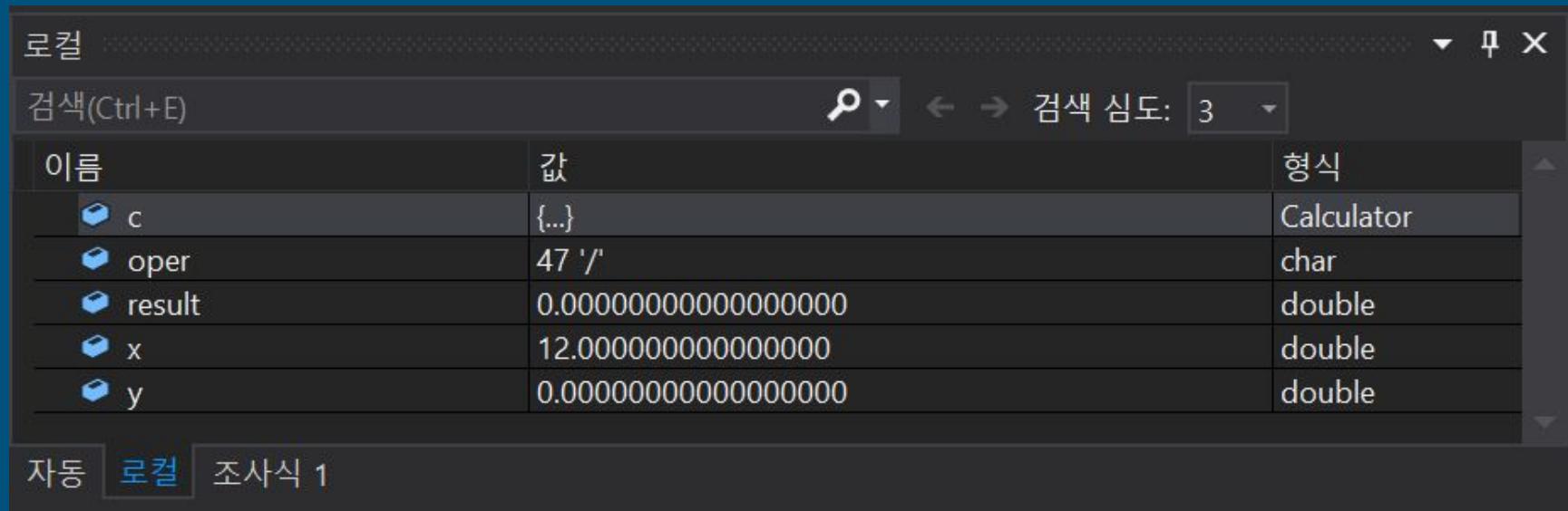
Whenever you debug your code, you may notice that some new windows appear. These windows can assist your debugging experience.

Take a look at the **Autos** window. The Autos window shows you the current values of variables used at least three lines before and up to the current line.

To see all of the variables from that function, switch to the **Locals** window. You can actually modify the values of these variables while debugging, to see what effect they would have on the program.

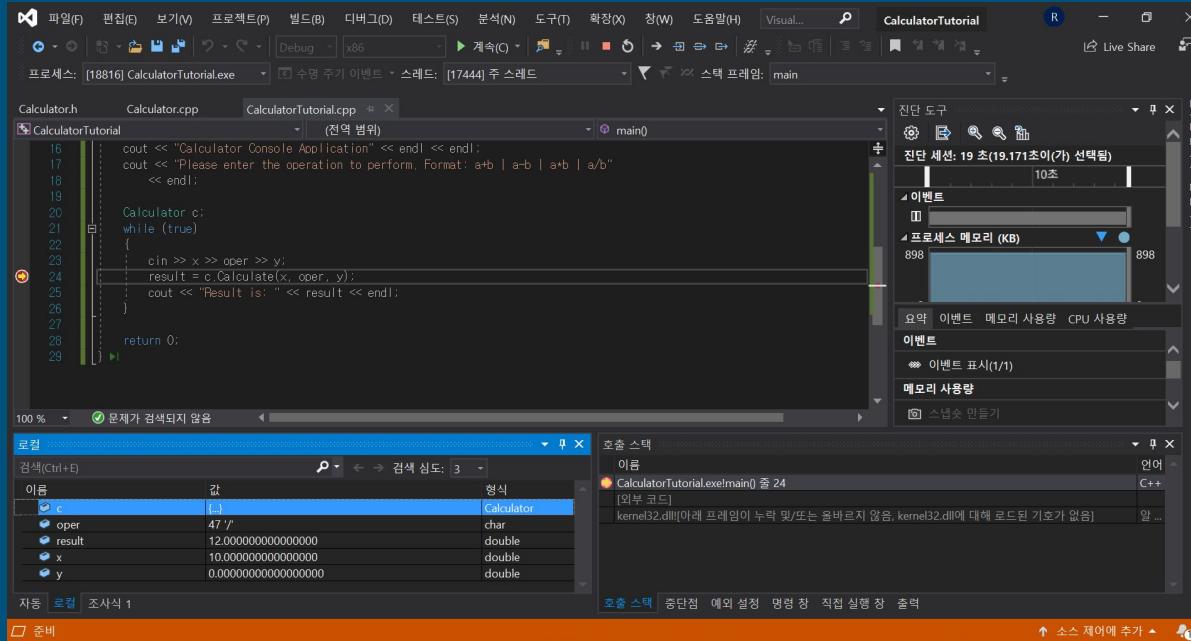
# Your first C++ console project - Useful windows in the debugger

---



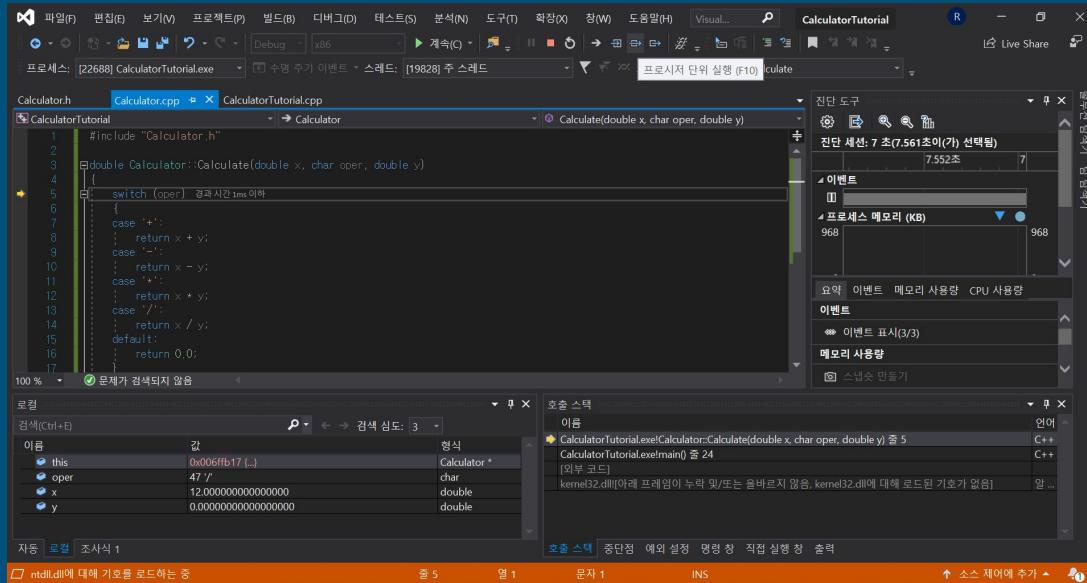
# Your first C++ console project - To continue debugging

1. The yellow line on the left shows the current point of execution. The current line calls Calculate, so **press F11** to Step Into the function. You'll find yourself in the body of the Calculate function.



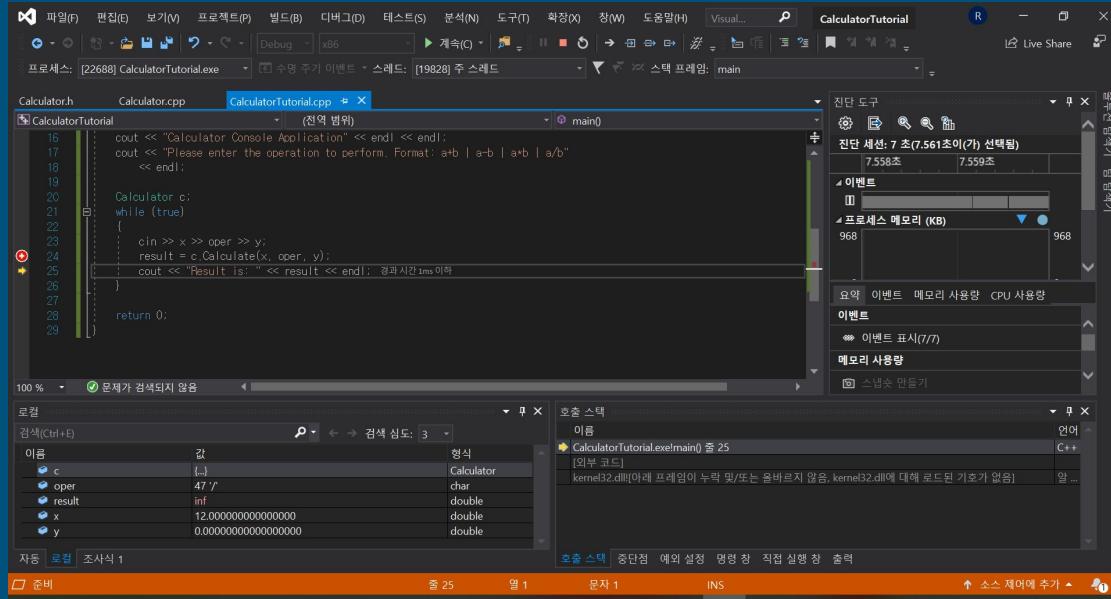
# Your first C++ console project - To continue debugging

2. Now that the point of execution is at the start of the Calculate function, press F10 to move to the next line in the program's execution. F10 is also known as Step Over. You can use Step Over to move from line to line, without delving into the details of what is occurring in each part of the line.



# Your first C++ console project - To continue debugging

3. Continue using **F10** to Step Over each line until you get back to the main() function in the other file, and stop on the cout line. You'll see its value is listed as "inf", which doesn't look right, so let's fix it.



# Your first C++ console project - To fix the "divide by zero" error

1. Edit the code to match **CalculatorTutorial3.cpp** in github.

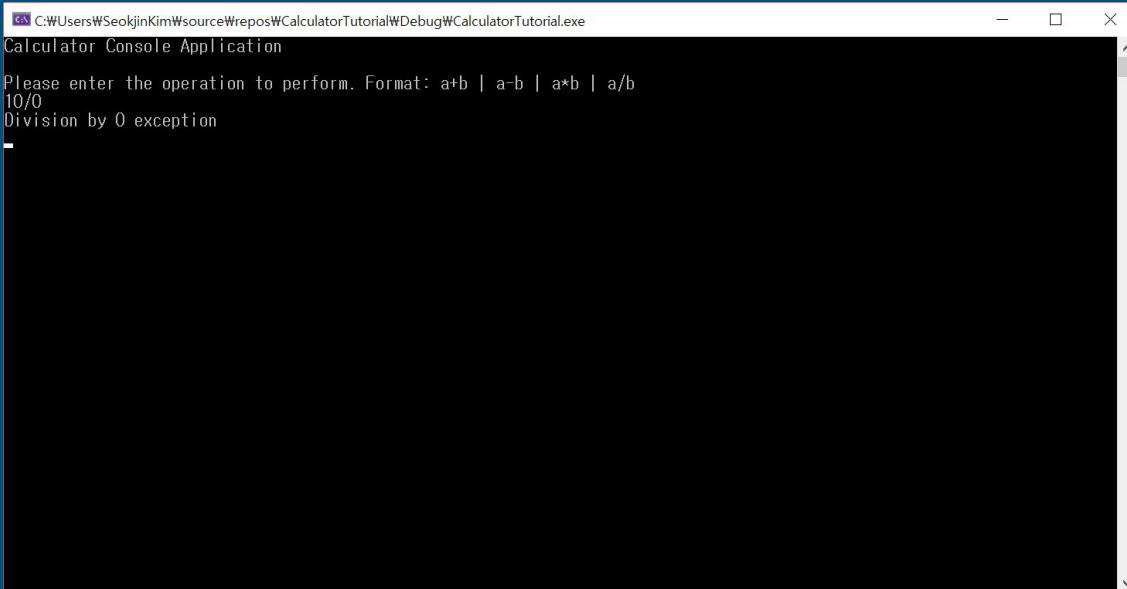
The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "CalculatorTutorial". The Solution Explorer on the right shows a single project named "CalculatorTutorial" with files like "Calculator.h", "Calculator.cpp", and "CalculatorTutorial.cpp". The "CalculatorTutorial.cpp" file is open in the main editor window, displaying C++ code for a calculator application. The code includes a function to handle division by zero. The Output window at the bottom shows the command prompt and the results of running the program, indicating it successfully compiled and ran without errors.

```
파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H) Visual... CalculatorTutorial R - ×
Calculator.h Calculator.cpp CalculatorTutorial.cpp x86 Debug 로컬 Windows 디버거 Live Share
CalculatorTutorial
Calculator.h
Calculator.cpp
CalculatorTutorial.cpp
main()
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
Calculator c;
while (true)
{
    cin >> x >> oper >> y;
    if (oper == '/' && y == 0)
    {
        cout << "Division by 0 exception" << endl;
        continue;
    }
}
문제가 검색되지 않음
100 %
문제가 검색되지 않음
출력
디버그
'CalculatorTutorial.exe' [Win32]: 'C:\Windows\SysWOW64\cruntime140.dll' 을(를) 로드했습니다.
'CalculatorTutorial.exe' [Win32]: 'C:\Windows\SysWOW64\crtbased.dll' 을(를) 로드했습니다.
0x50fc 스레드가 종료되었습니다.(코드: 0 (0x0)).
0x4d7c 스레드가 종료되었습니다.(코드: -1073741510 (0xc000013a)).
0x304c 스레드가 종료되었습니다.(코드: -1073741510 (0xc000013a)).
0x497c 스레드가 종료되었습니다.(코드: -1073741510 (0xc000013a)).
[15696] CalculatorTutorial.exe 프로그램이 종료되었습니다.(코드: -1073741510 (0xc000013a)).
```

# Your first C++ console project - To fix the "divide by zero" error

---

2. Now you can see that error message is printed when entered 10/0!



A screenshot of a Windows Command Prompt window titled "Calculator Console Application". The window shows the following text:  
C:\Users\SeokjinKim\source\repos\CalculatorTutorial\Debug\CalculatorTutorial.exe  
Please enter the operation to perform. Format: a+b | a-b | a\*b | a/b  
10/0  
Division by 0 exception

# How to handle external library in Visual Studio

# How to handle external library in VS

1. Create a new console project named 'Parrot'.
2. Edit the **Parrot.cpp** to match **Parrot.cpp** in github.
3. You'll see a red squiggle appear under #include and GetString().

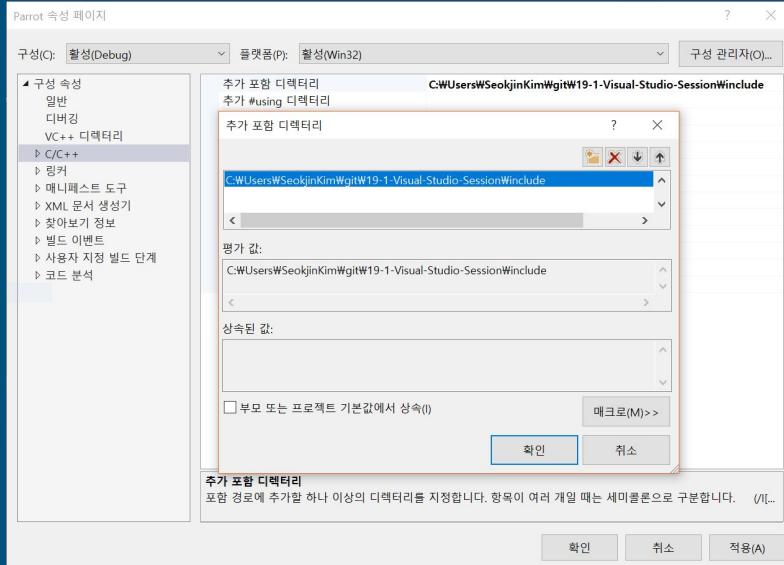
The screenshot shows the Visual Studio IDE interface with the following details:

- File Menu:** 파일(F), 편집(E), 보기(V), 프로젝트(P), 빌드(B), 디버그(D), 테스트(S), 분석(N), 도구(T), 확장(X), 창(W), 도움말(H).
- Toolbar:** Standard toolbar icons.
- Status Bar:** 100%, 2, 0, 출력, 출력 보기 선택(S):.
- Code Editor:** The file "Parrot.cpp" is open. The code includes a "#include <nowich.h>" directive which has a red squiggle under it, indicating a warning or error. The code also contains a call to "GetString()" which has a red squiggle under it.
- Solution Explorer:** Shows the solution "Parrot" with one project "Parrot" containing files "Parrot.cpp" and "헤더 파일".
- Toolbars:** Standard Visual Studio toolbars.
- Task List:** Shows "솔루션 탐색기" (Solution Explorer) and "팀 탐색기" (Team Explorer).
- Output Window:** Shows the output of the build process.

# How to handle external library in VS

## 4. You must enter the path to the header file

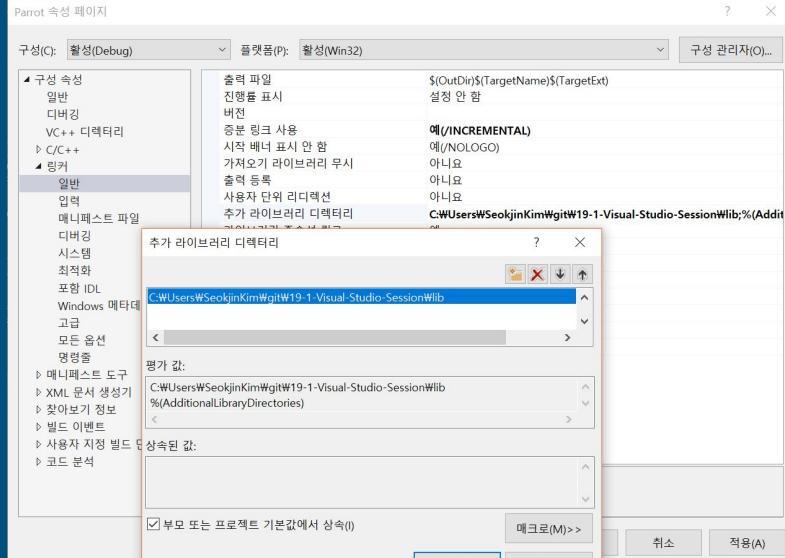
Solution Explorer >> Right click on the project >> setting >> C/C++ >> general >> additional include directories



# How to handle external library in VS

## 5. You must enter the path to the static library

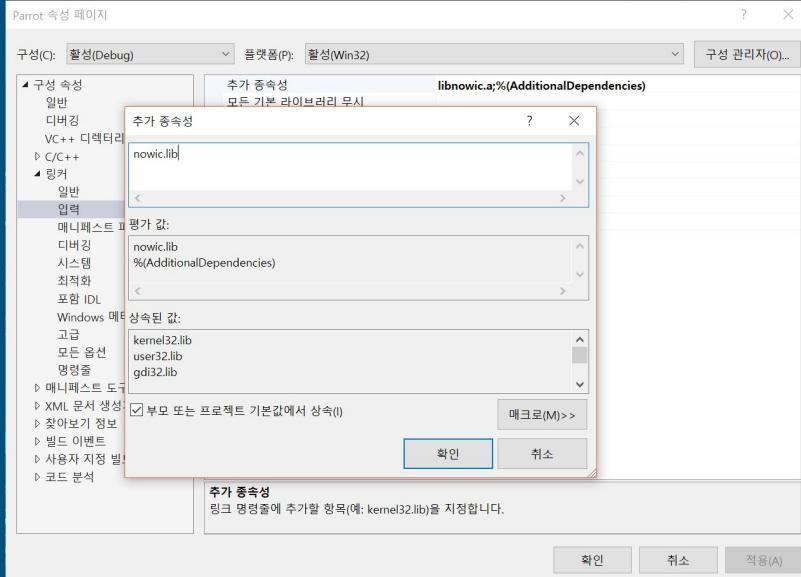
Solution Explorer >> Right click on the project >> setting >> Linker >> general >> additional library directories



# How to handle external library in VS

## 5. You must enter the name of static library

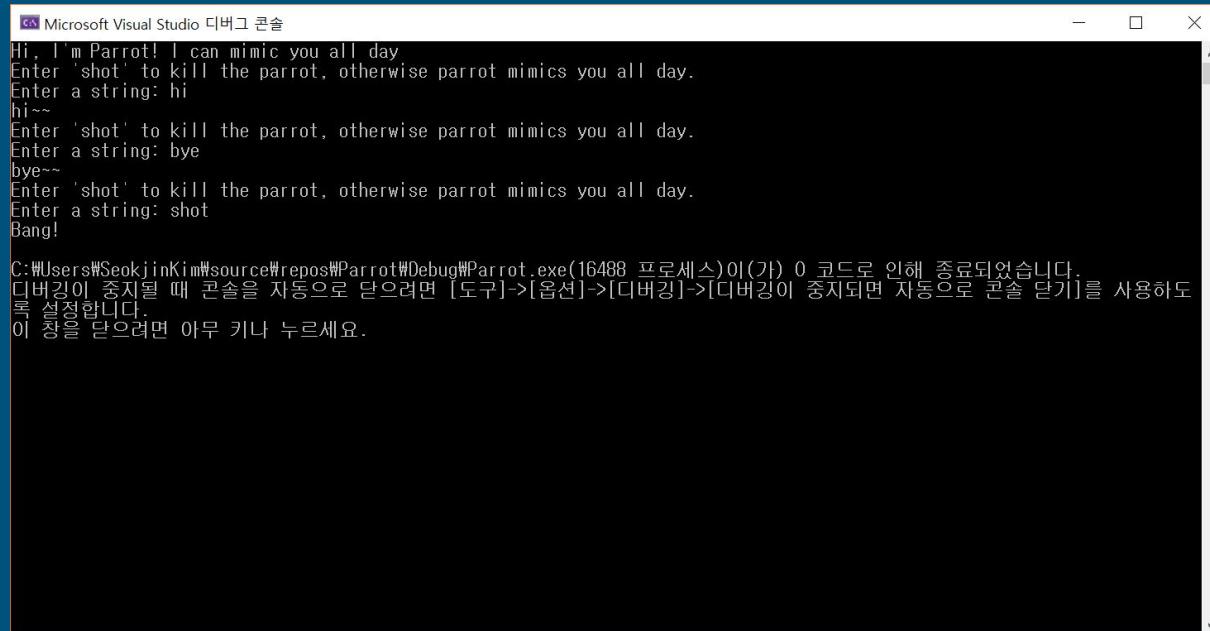
Solution Explorer >> Right click on the project >> setting >> Linker >> general >> additional dependencies



# How to handle external library in VS

---

## 7. Build and test Parrot!



Microsoft Visual Studio 디버그 콘솔

```
Hi, I'm Parrot! I can mimic you all day.
Enter 'shot' to kill the parrot, otherwise parrot mimics you all day.
Enter a string: hi
hi~~
Enter 'shot' to kill the parrot, otherwise parrot mimics you all day.
Enter a string: bye
bye~~
Enter 'shot' to kill the parrot, otherwise parrot mimics you all day.
Enter a string: shot
Bang!
C:\Users\SeokjinKim\source\repos\Parrot\Debug\Parrot.exe(16488 프로세스)이(가) 0 코드로 인해 종료되었습니다.
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구]->[옵션]->[디버깅]->[디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다.
이 항목을 닫으려면 아무 키나 누르세요.
```

# References

---

## **Create a C++ console app project**

<https://docs.microsoft.com/ko-kr/cpp/get-started/tutorial-console-cpp?view=vs-2019>

## **Visual Studio IDE Introduction**

<https://docs.microsoft.com/ko-kr/visualstudio/get-started/visual-studio-ide?view=vs-2019>

## **How to handle external library in Visual Studio**

[https://wnsgml972.github.io/setting/dll\\_lib.html](https://wnsgml972.github.io/setting/dll_lib.html)