

CS2311 Computer Programming

Labo8:

Introduction to
Visual Studio 2019 Debugger

Debugger

- What is a debugger?
 - ▶ A tool for programmer to
 - ▶ Trace the execution path of a program
 - ▶ Step through the code line by line and investigate the value of variables at each step
- When you need a debugger?
 - ▶ When a program does not act as the programmer expected, the debugger can help to find out the logical bug

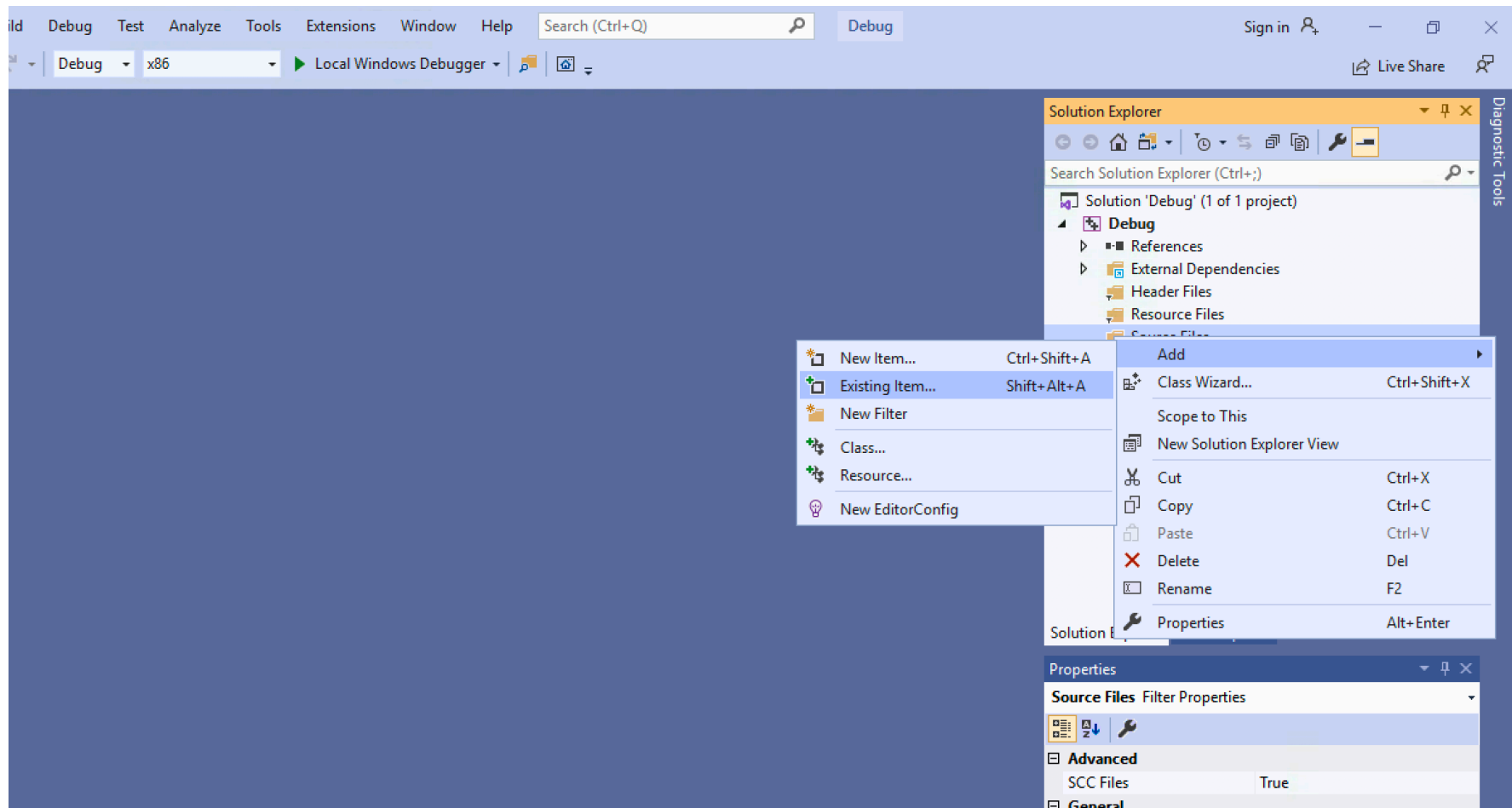
Outlines

- In this lab you will learn how to
 1. Trace a program
 2. Display the value of variable
- Finish the exercises

Lab Exercise

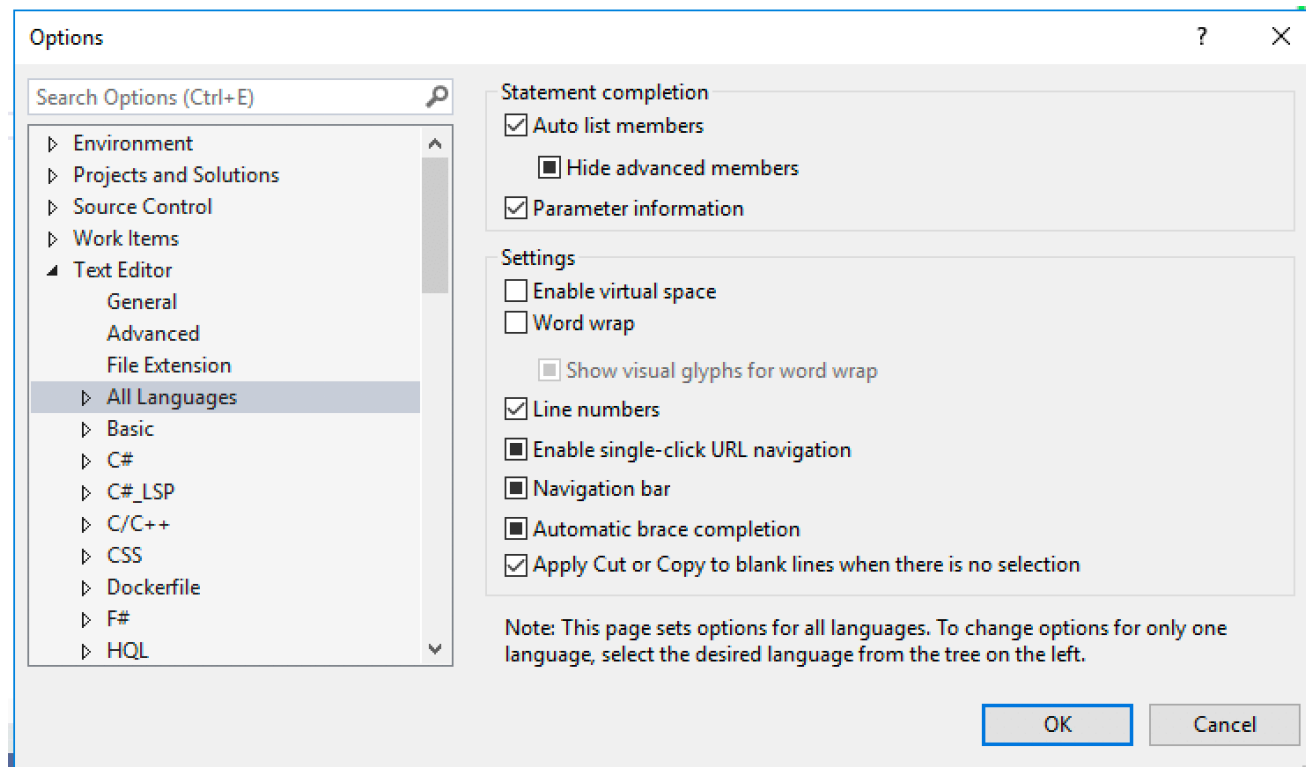
- Debug a sample program (**debug1.cpp**)
- Steps:
 - ▶ Create a new project in visual studio
 - ▶ Download the **debug1.cpp** from backboard and save it to the project folder
 - ▶ In visual studio, add an existing item to the project

Add the sample program to the project



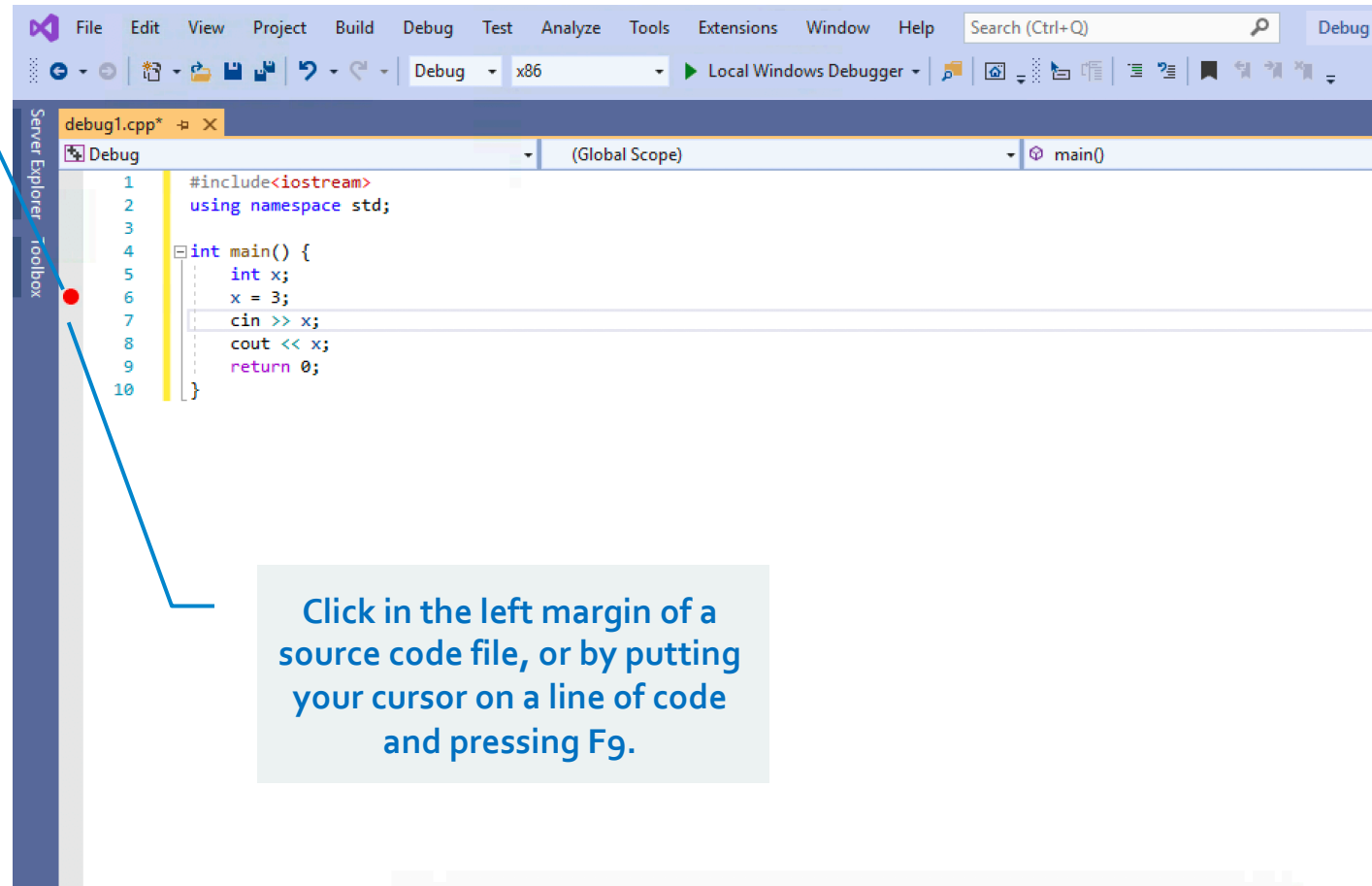
Tip: Display Line Numbers in Visual Studio

- To display line numbers in code
 - ▶ On the menu bar, choose **Tools, Options**. Expand the **Text Editor** node, and then select either the node for the language you are using, or **All Languages** to turn on line numbers in all languages. Or, you can type **line number** in the **Quick Launch** box.
 - ▶ Select the **Line numbers** checkbox.



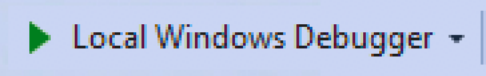
Set a breakpoint

The breakpoint appears as a red dot in the left margin.

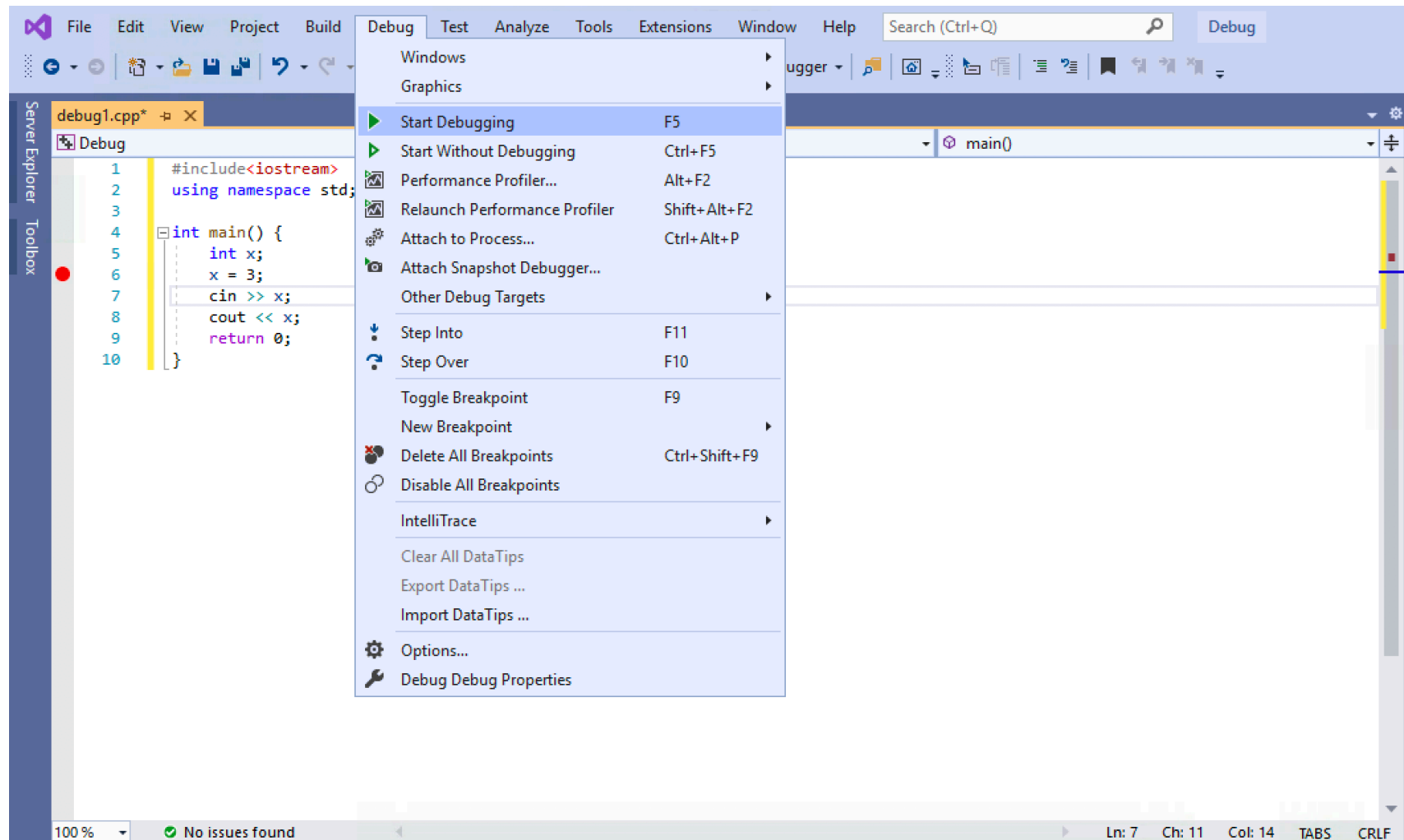


Click in the left margin of a source code file, or by putting your cursor on a line of code and pressing F9.

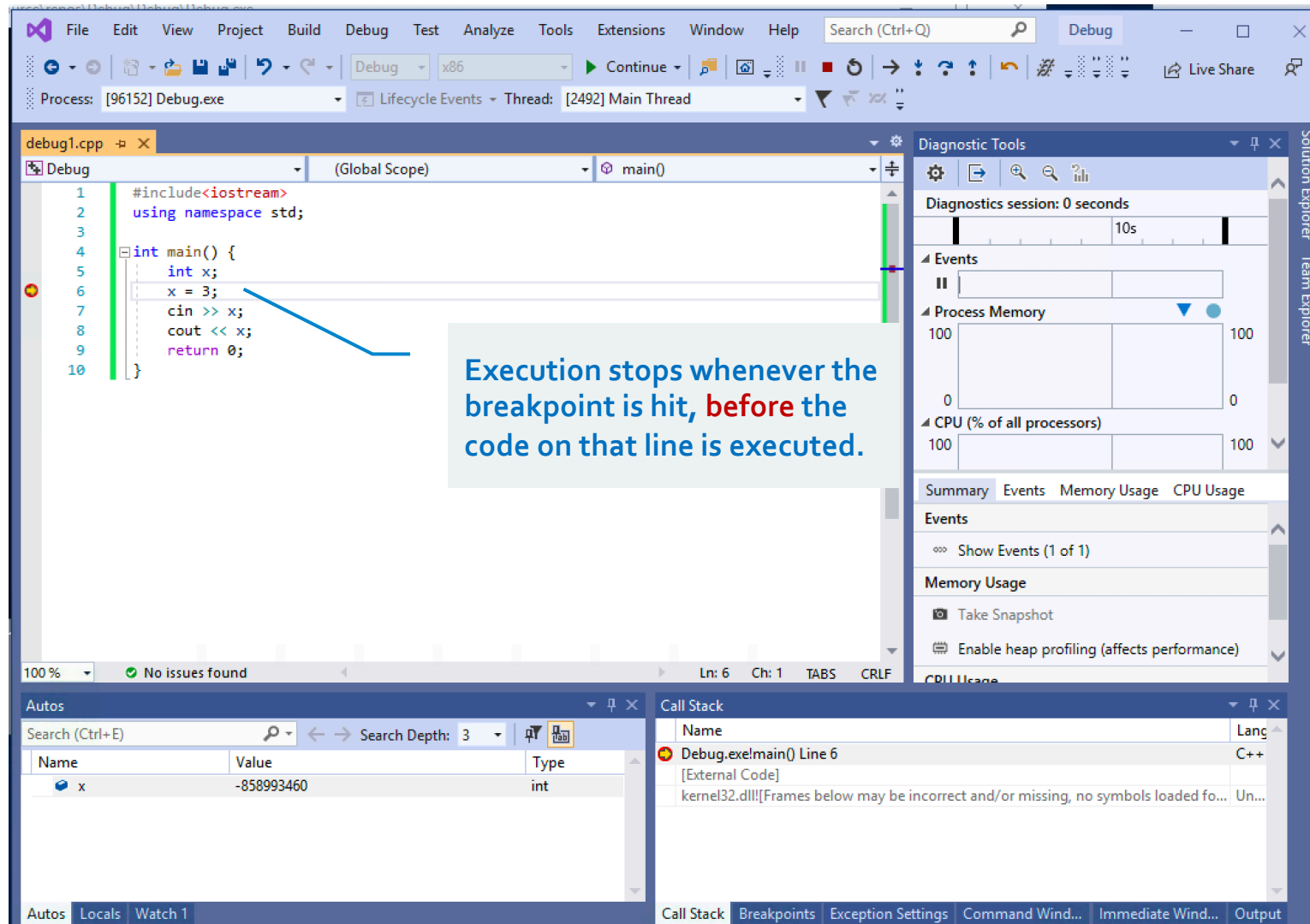
Steps to invoke the debugger

- Build/Rebuild the solution
- Start the debugger
 - ▶ Start a debugging session using **F5** (**Debug > Start Debugging**). This command starts your app with the debugger attached.
 - ▶ The green arrow on the tool bar also starts the debugger (same as **F5**). 
- The program execution should stop at the line "**x=3**"

Start the debugger (menu bar)



Execution stop at x=3



Step through a program

- **Step over (F10):**

- ▶ Execute the current statement and stop at the next statement.
- ▶ If the current statement is a function call, the function will be executed.

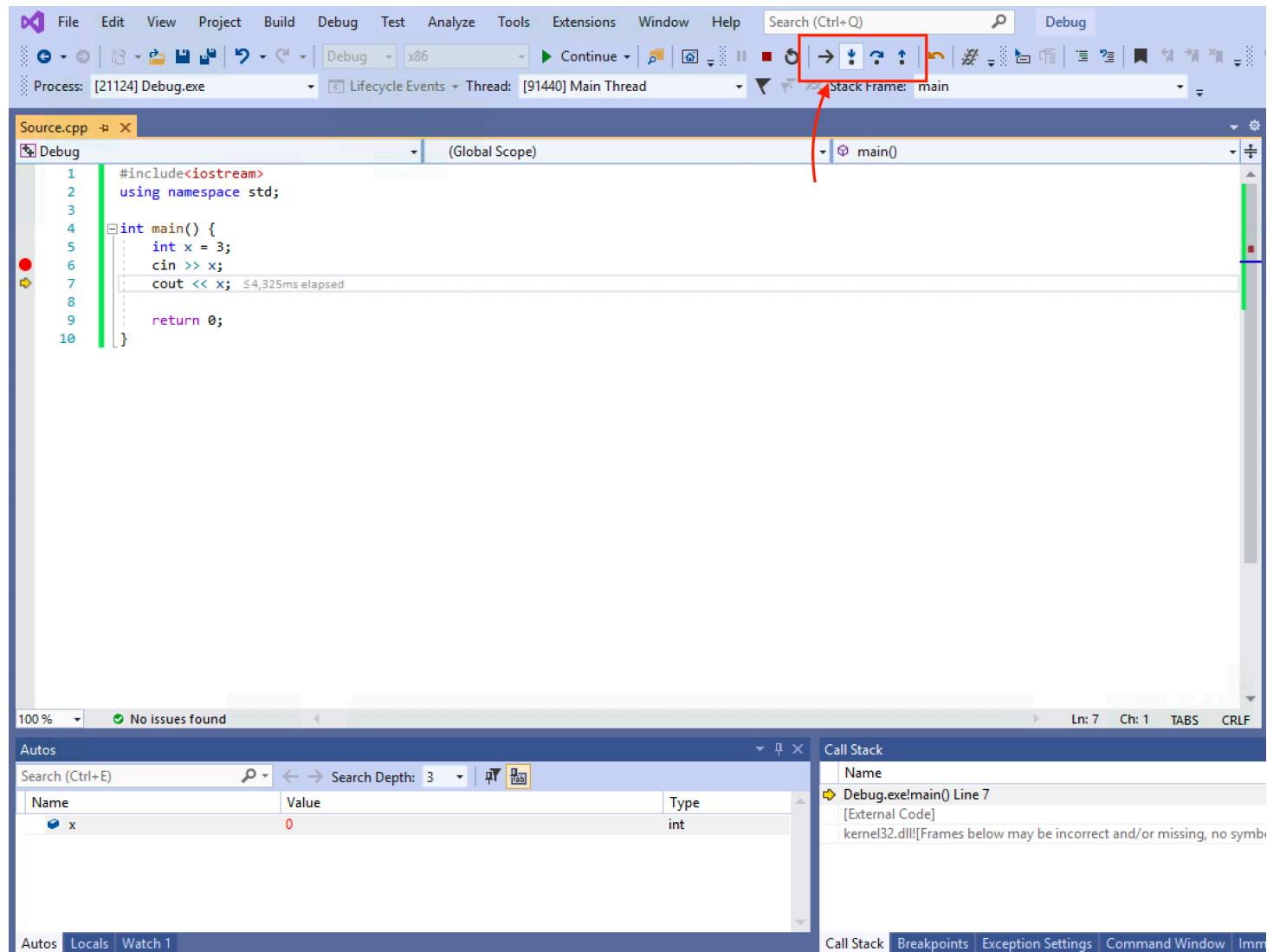
- **Step Into (F11):**

- ▶ If the current statement is a function call, go inside the function body and stop at the first statement.

- **Step out (Shift+F11):**

- ▶ Finish the execution of current function and stop at the point where the function is called.

Step into, Step over, Step out



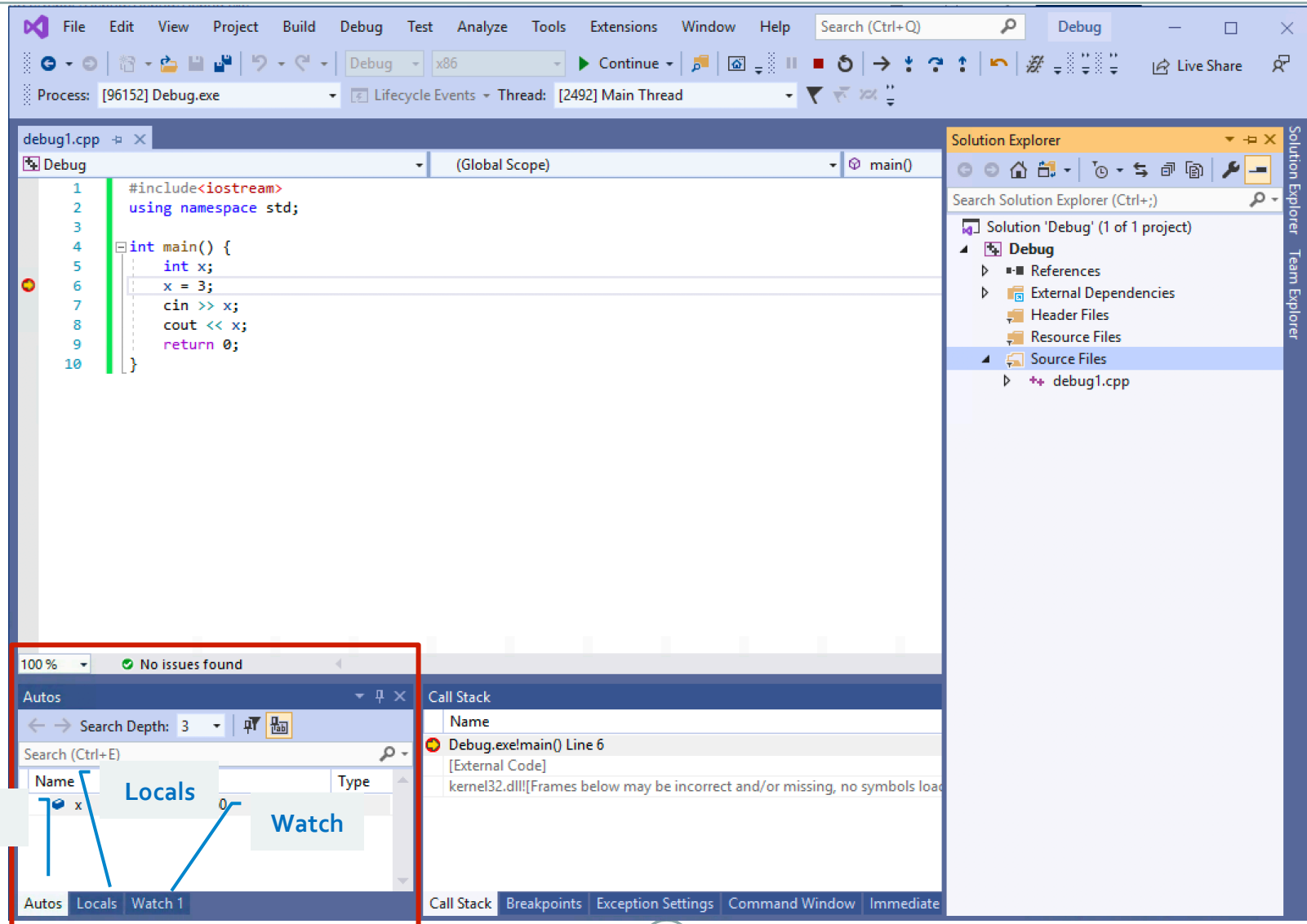
Summary of short-cut key

Key	Function
Ctl+alt+F7	Rebuild the project
F5	Start Debugging
Ctl+F5	Start Without Debugging
Shift+F5	Stop Debugging
F9	Set Breakpoint
F10	Step Over
Ctl+F10	Run To Cursor
F11	Step Into
Shift+F11	Step Out

Display the value of a variable

- Value of variable can be found in
 - ▶ **Autos**: display related variables (selected by VS)
 - ▶ **Locals**: display local variable only
 - ▶ **Watch**_{1..N}: display user selected variable

Display the value of a variable



Example: $x = -858993460$

The screenshot shows the Visual Studio IDE with a C++ project named 'Debug'. The source file 'debug1.cpp' contains the following code:

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     int x;
6     x = 3;
7     cin >> x;
8     cout << x;
9     return 0;
10 }
```

The program is running in Debug mode. The 'Autos' window shows the current state of the variable `x`:

Name	Value	Type
x	-858993460	int

The 'Call Stack' window shows the current frame: 'Debug.exe!main() Line 6'. The 'Solution Explorer' shows the project structure with 'debug1.cpp' as the source file.

Example: $x=3$

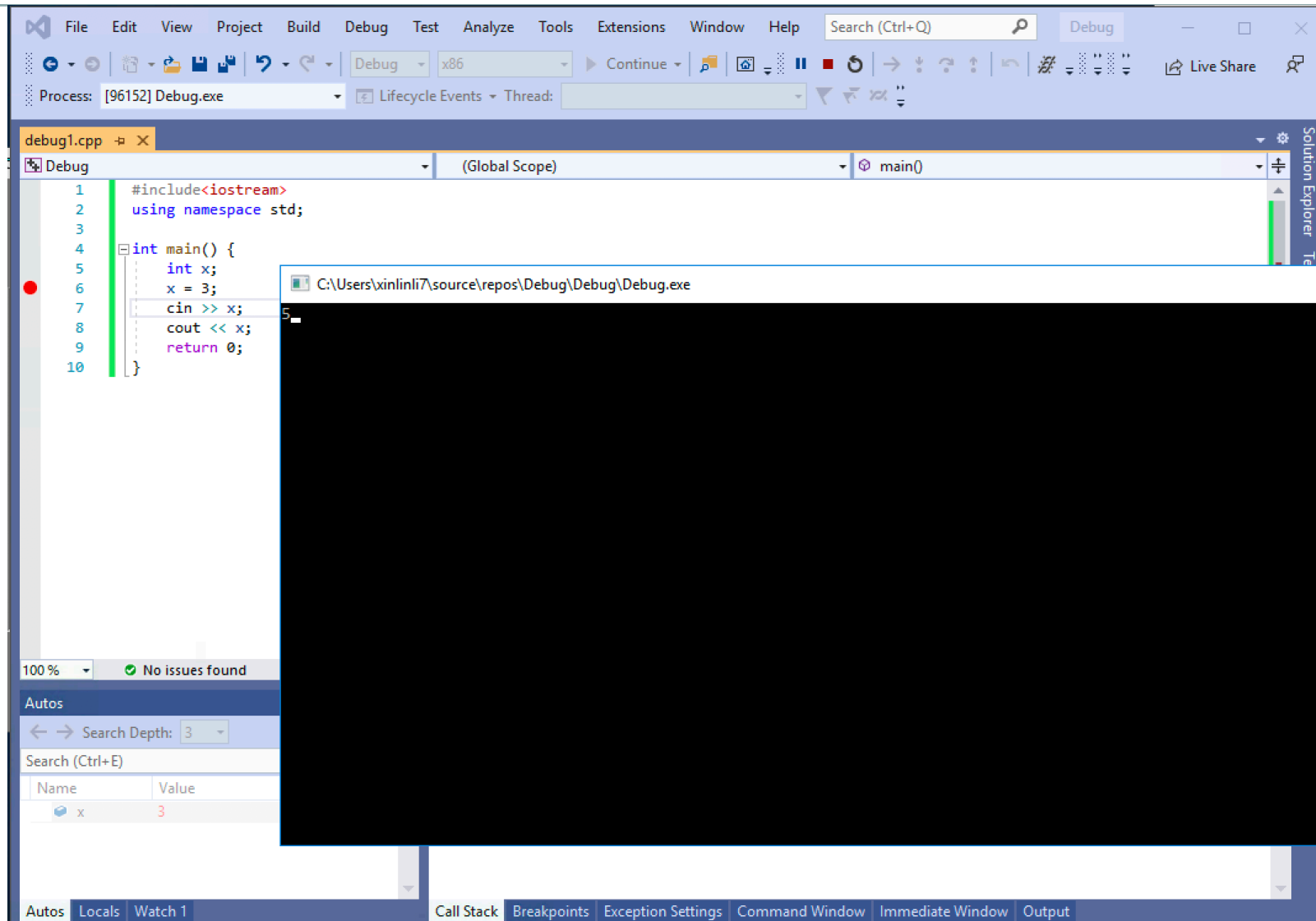
The screenshot shows the Visual Studio Code interface with a C++ program being debugged. The program is `debug1.cpp`, and it's running on a `Debug` process. The code is paused at line 7, where `x` is assigned the value 3. The Autos window shows the variable `x` with value 3 and type `int`. The Call Stack shows the current frame is `Debug.exe!main()` at line 7.

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     int x;
6     x = 3;
7     cin >> x;
8     cout << x;
9     return 0;
10 }
```

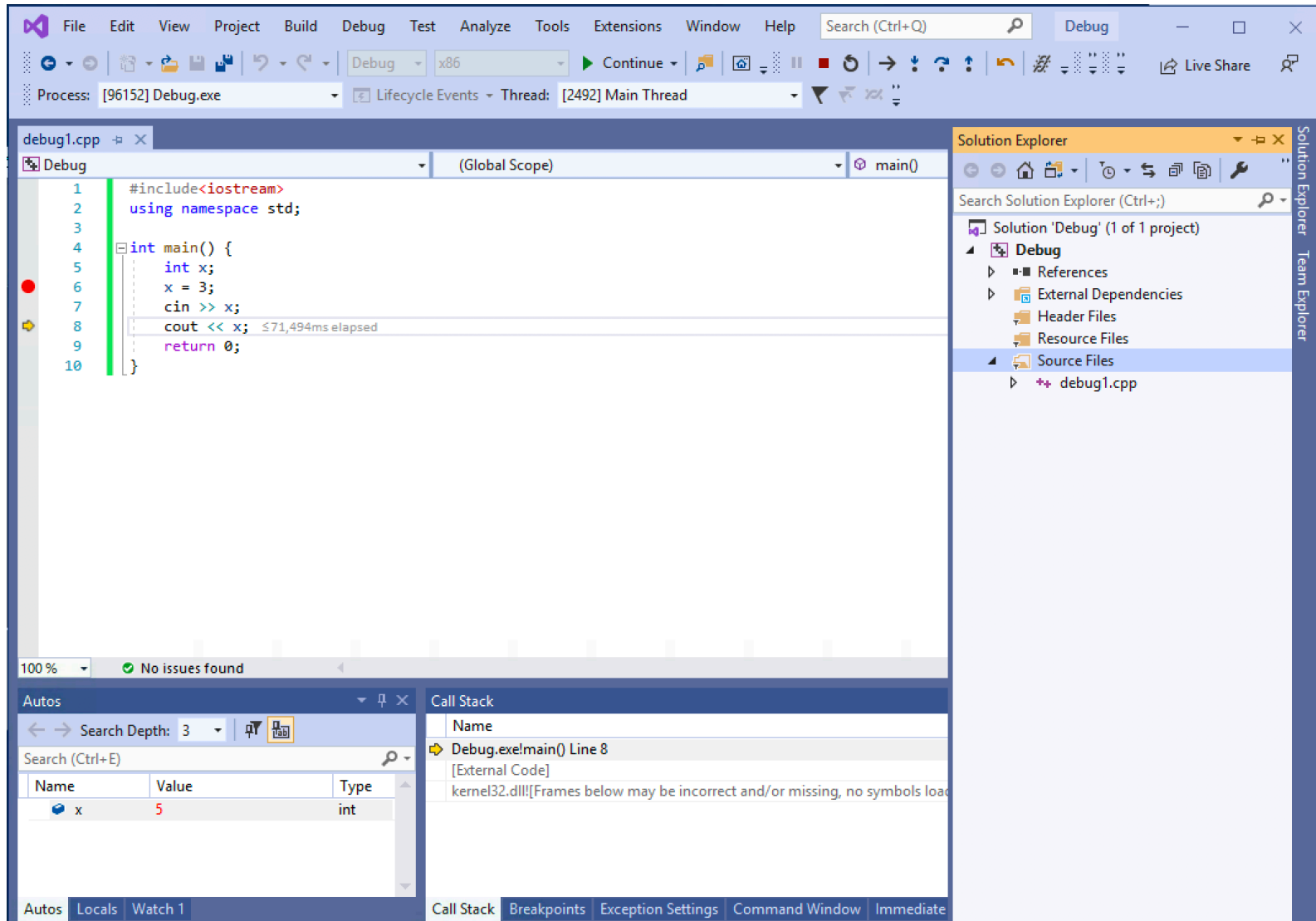
Name	Value	Type
x	3	int

Name	Lang
Debug.exe!main() Line 7	C++
[External Code]	
kernel32.dll![]	Un...

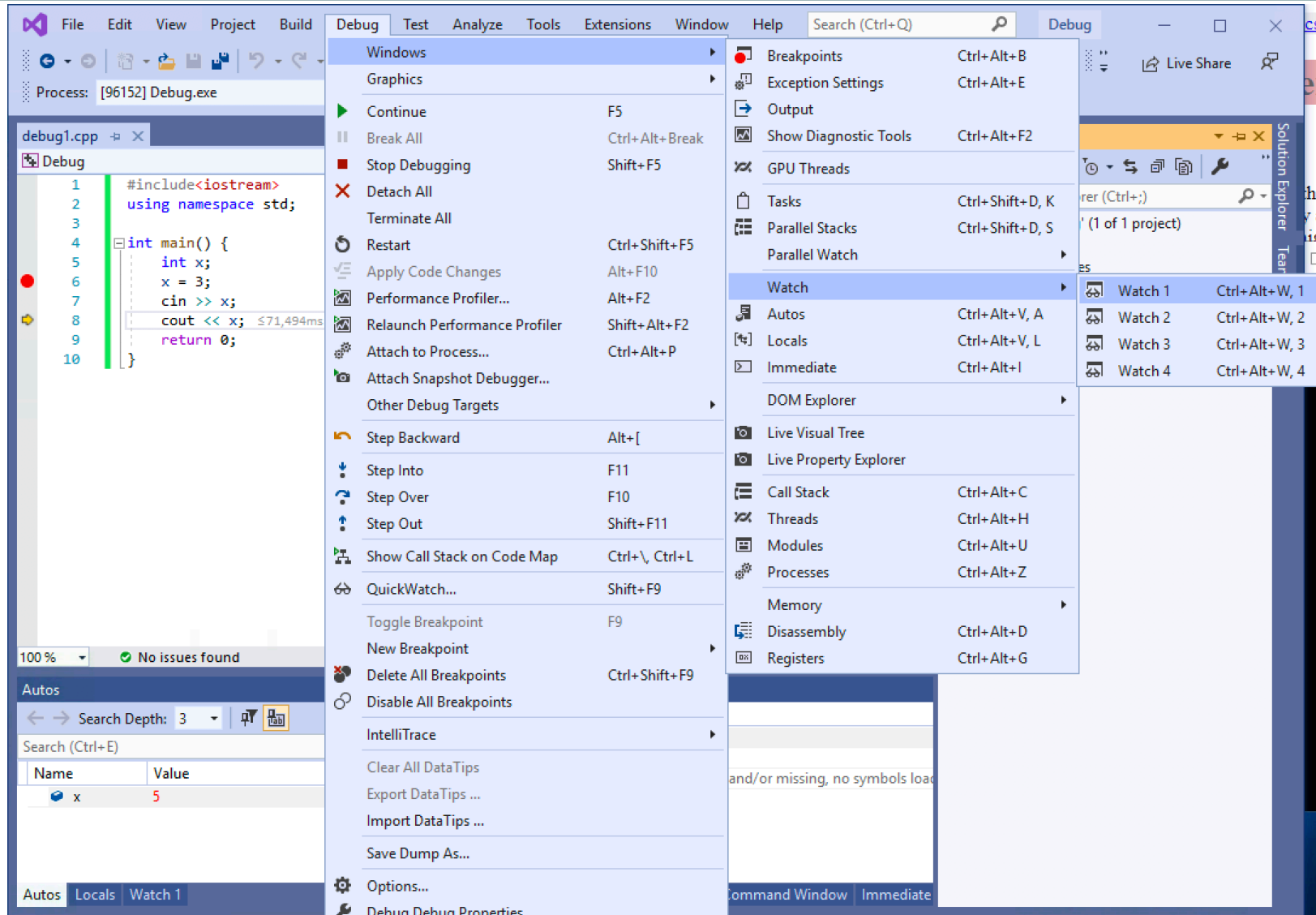
Example: input new **x** (e.g. 5)



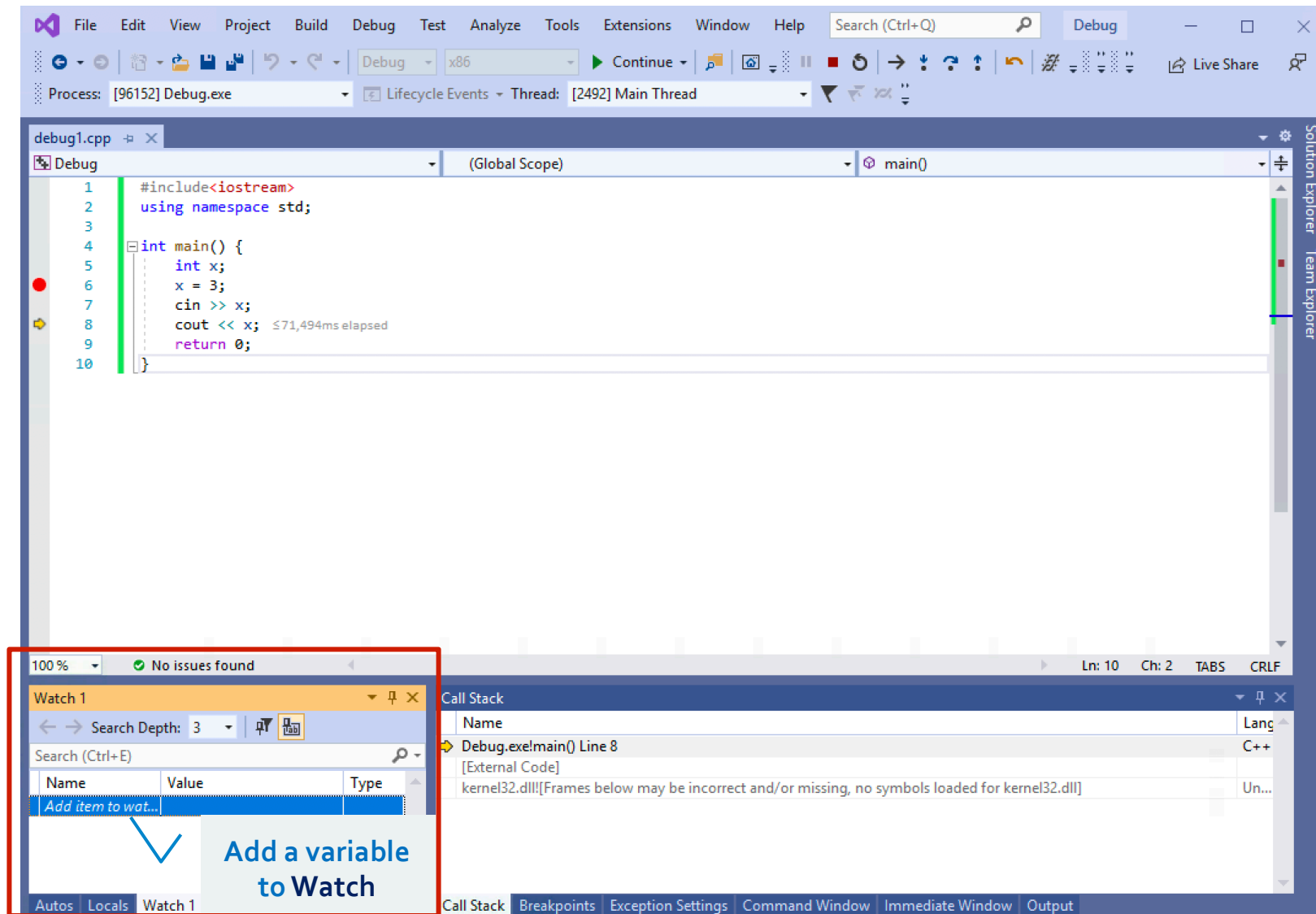
Example: $x=5$



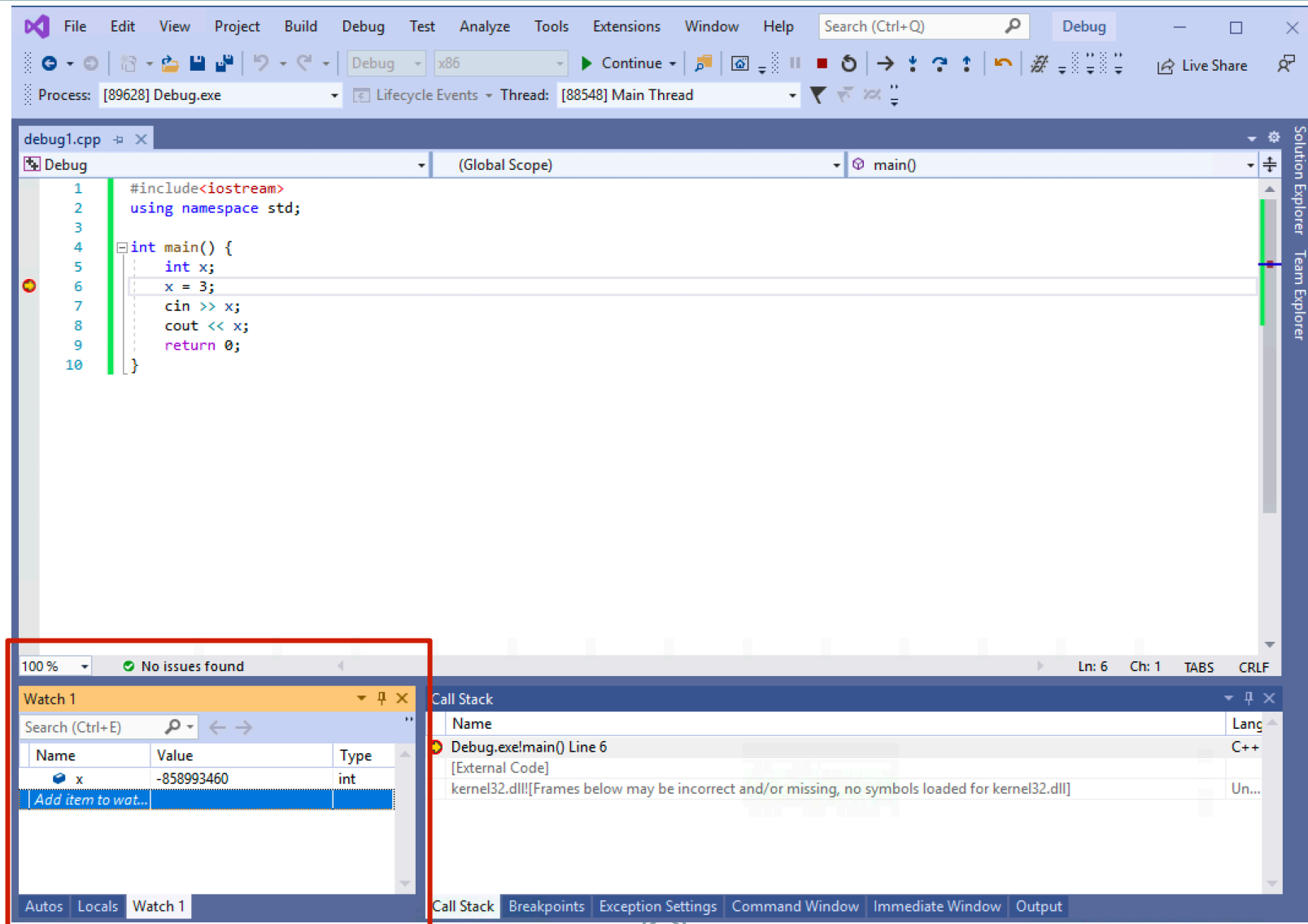
If you can't find the Watch windows, ...



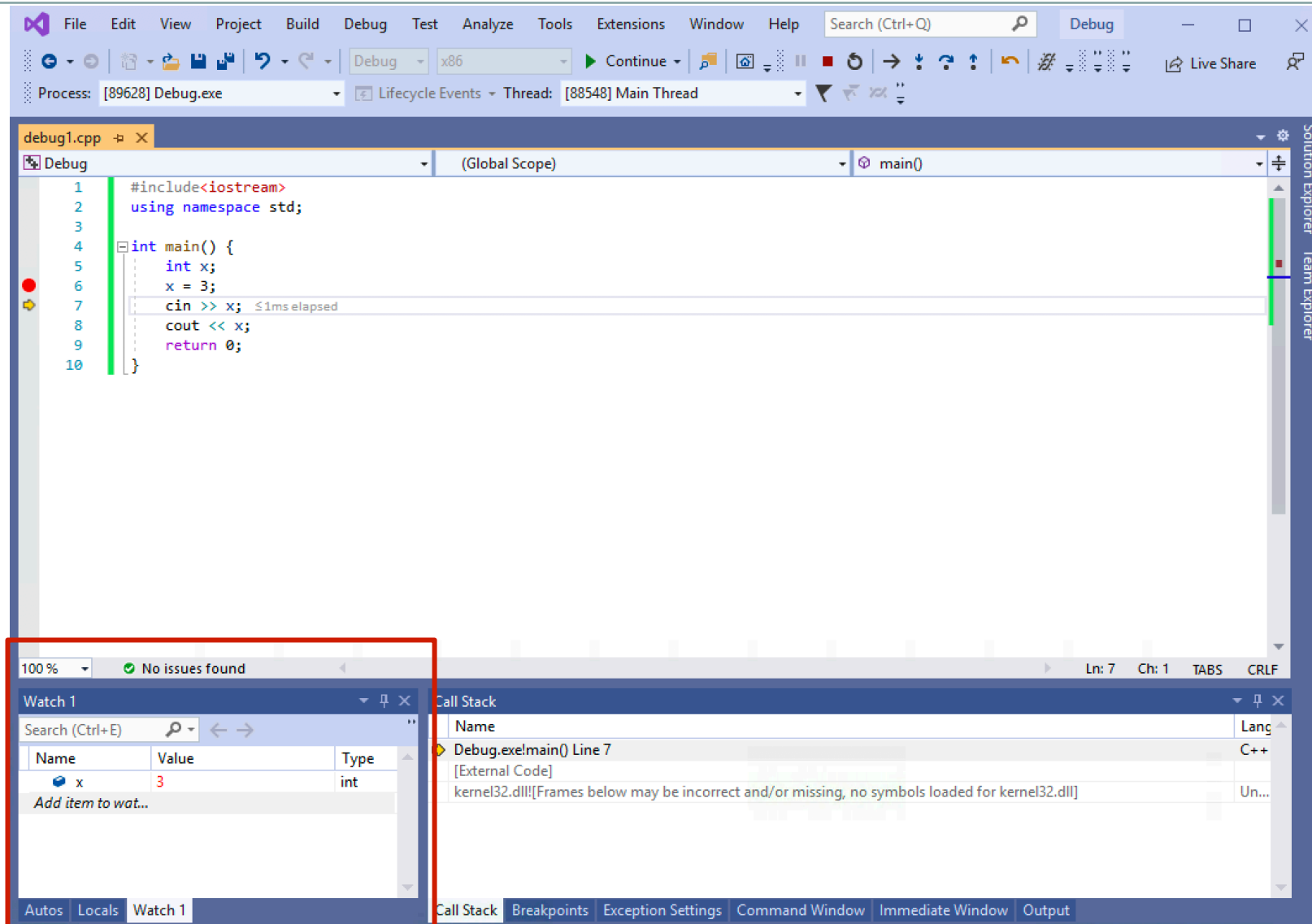
Add a variable to Watch



Add a variable to Watch (cont'd)

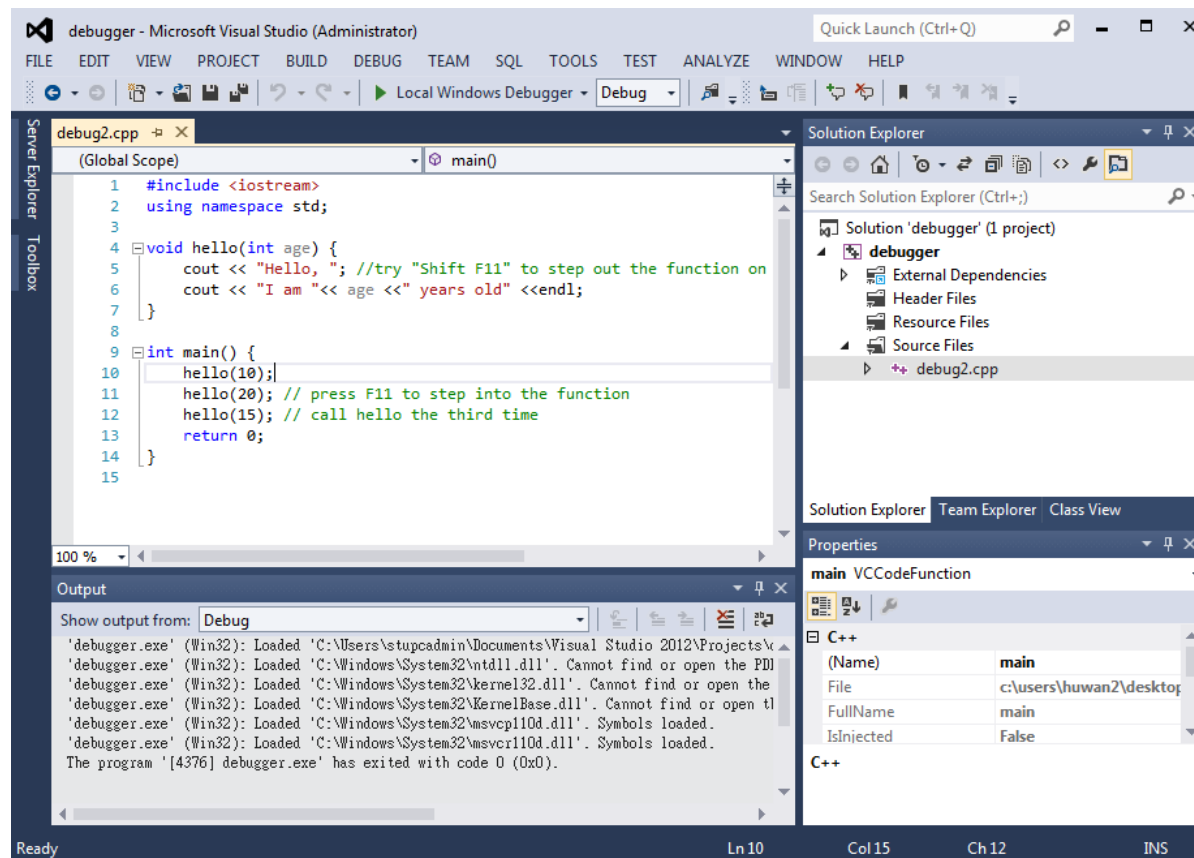


Add a variable to Watch (cont'd)



Exercise 1

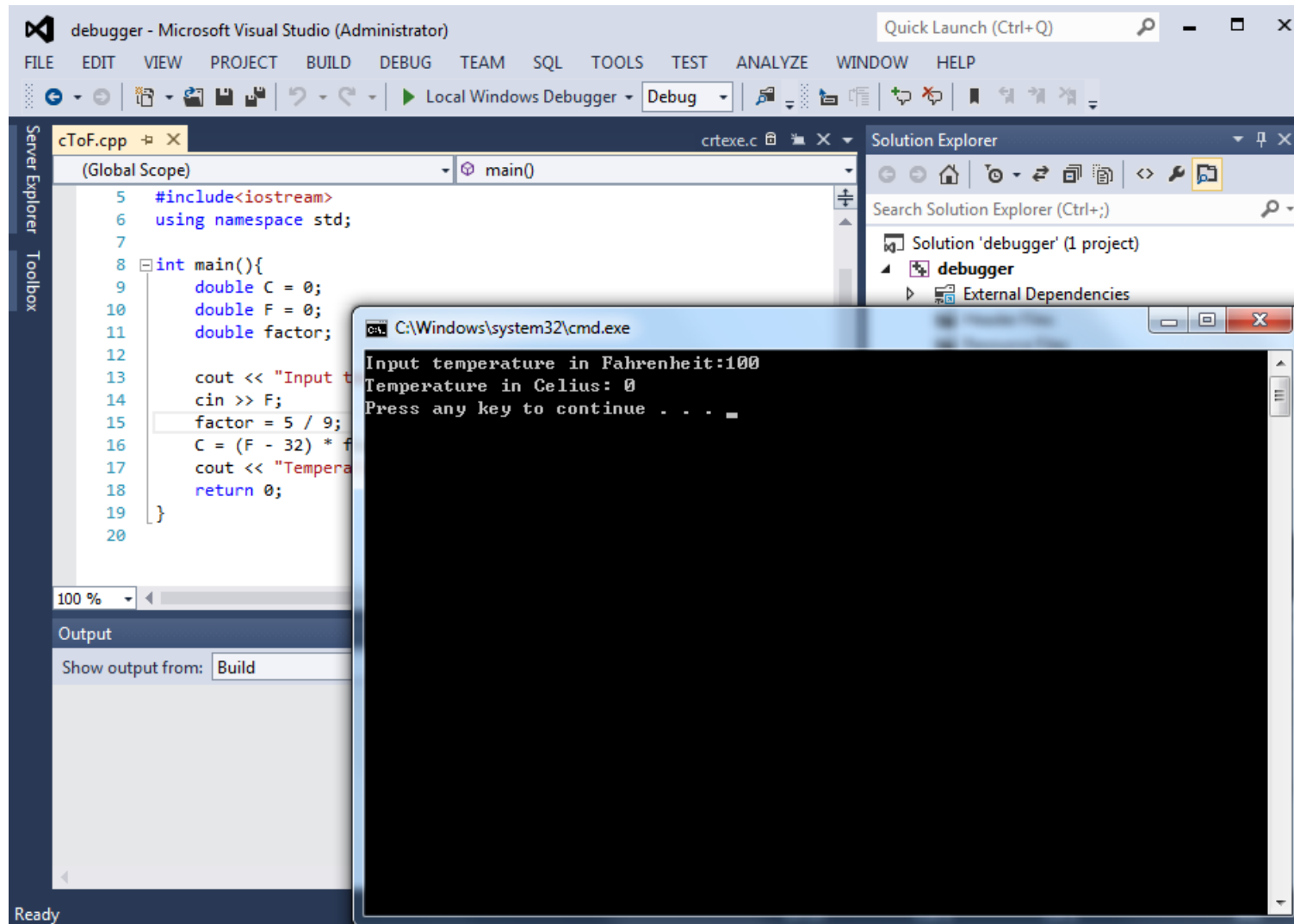
- Download the **debug2.cpp**
- Use **step over**, **step into** and **step out** to trace the program



Exercise 2

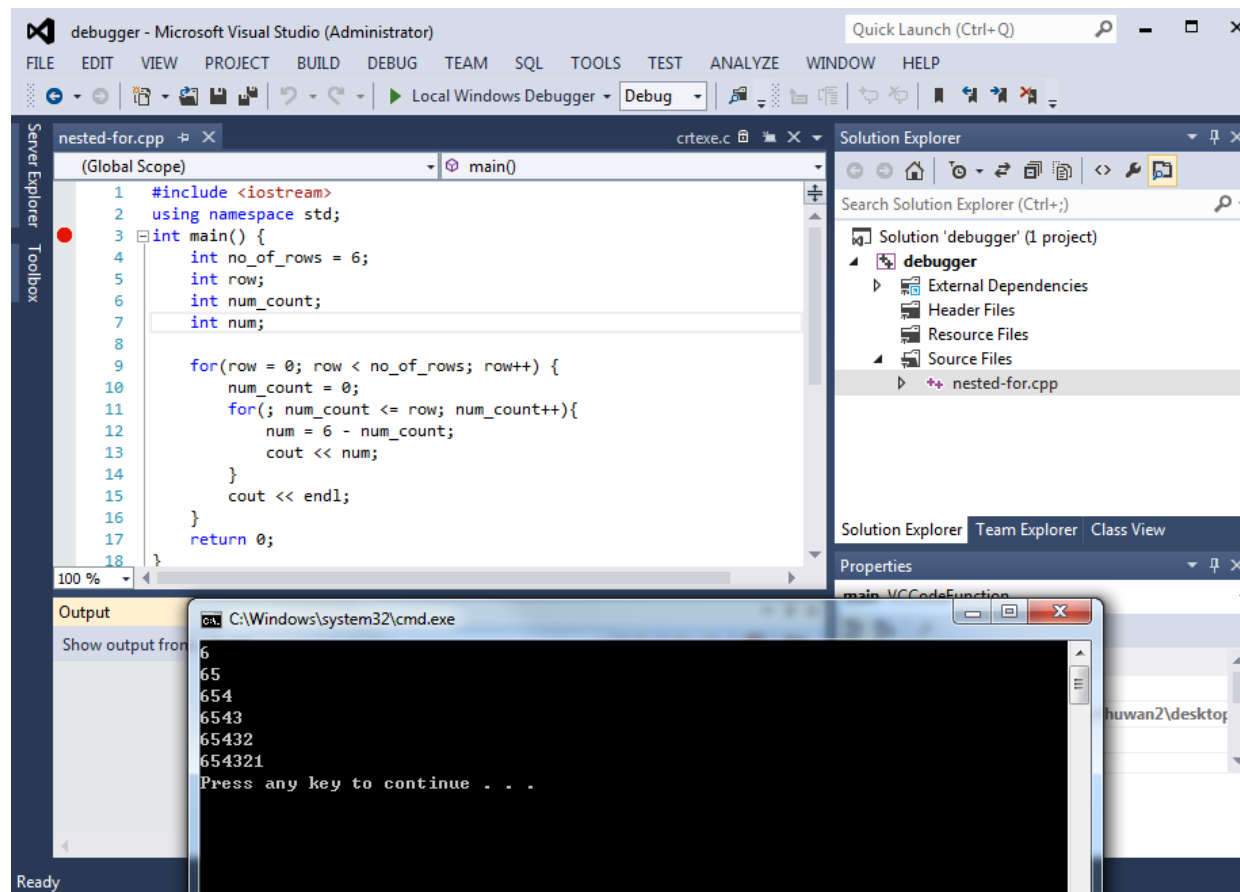
- Download the program **cToF.cpp**
 - ▶ The program accepts a real number in Fahrenheit (F) and output the temperature in Celsius®, where $C = (F - 32) * 5/9$.
 - ▶ Compile and execute the program.
 - ▶ **Type 100 as the input**
- Notice that the output of the program is **not correct** and the program contains **logical errors**.
- Try to locate and debug the program using the VS debugger
 - ▶ Hint: what is the value of $5/9$?

Exercise 2: Wrong Output



Exercise 3

- Download **nestedfor.cpp** and run inside a debugger to trace the execution of the program



Exercise 4

- Download **sumArray.cpp**, correct the logical errors in the program with the use of the debugger

