

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

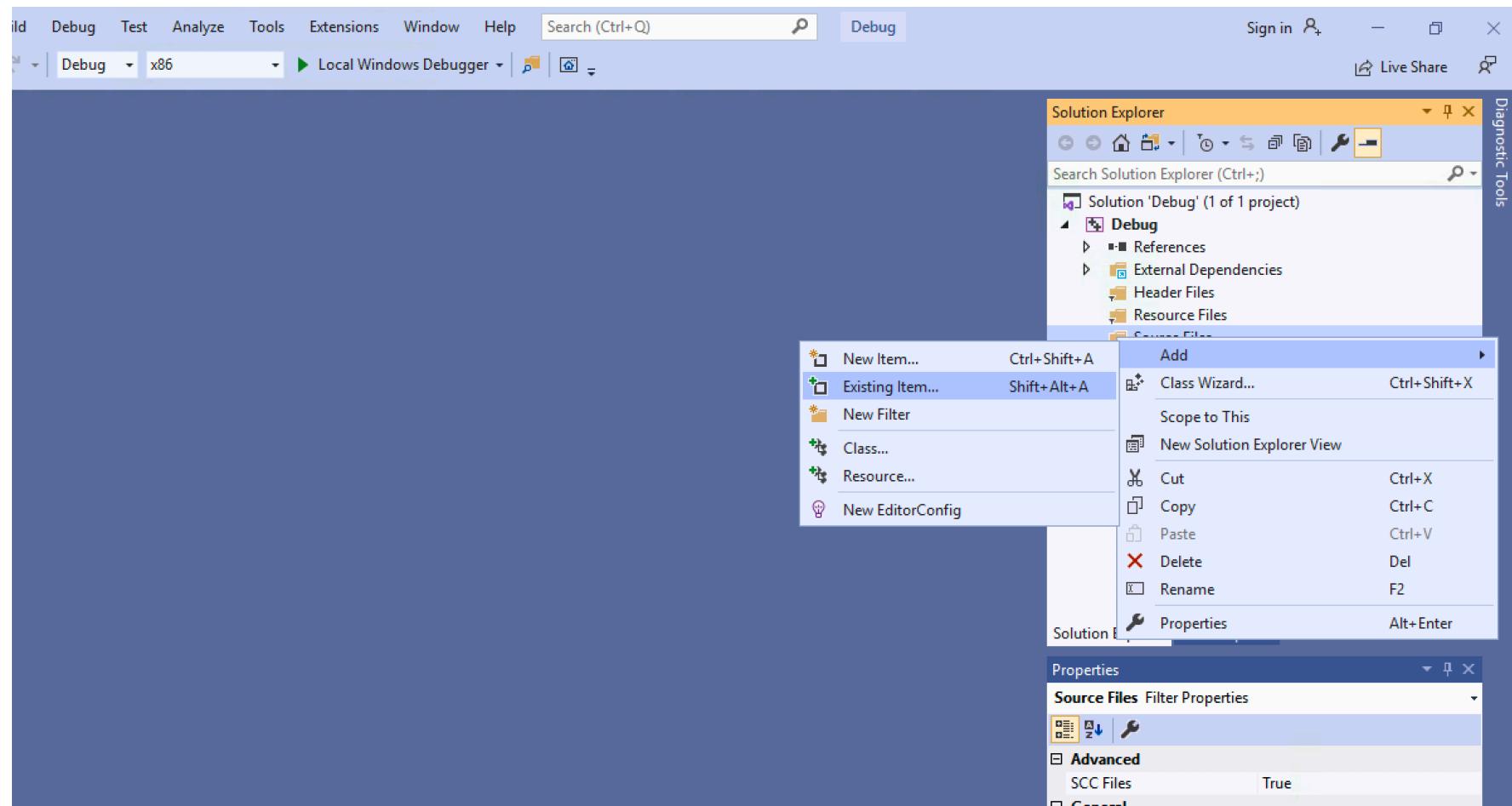
Outlines

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

Lab Exercise

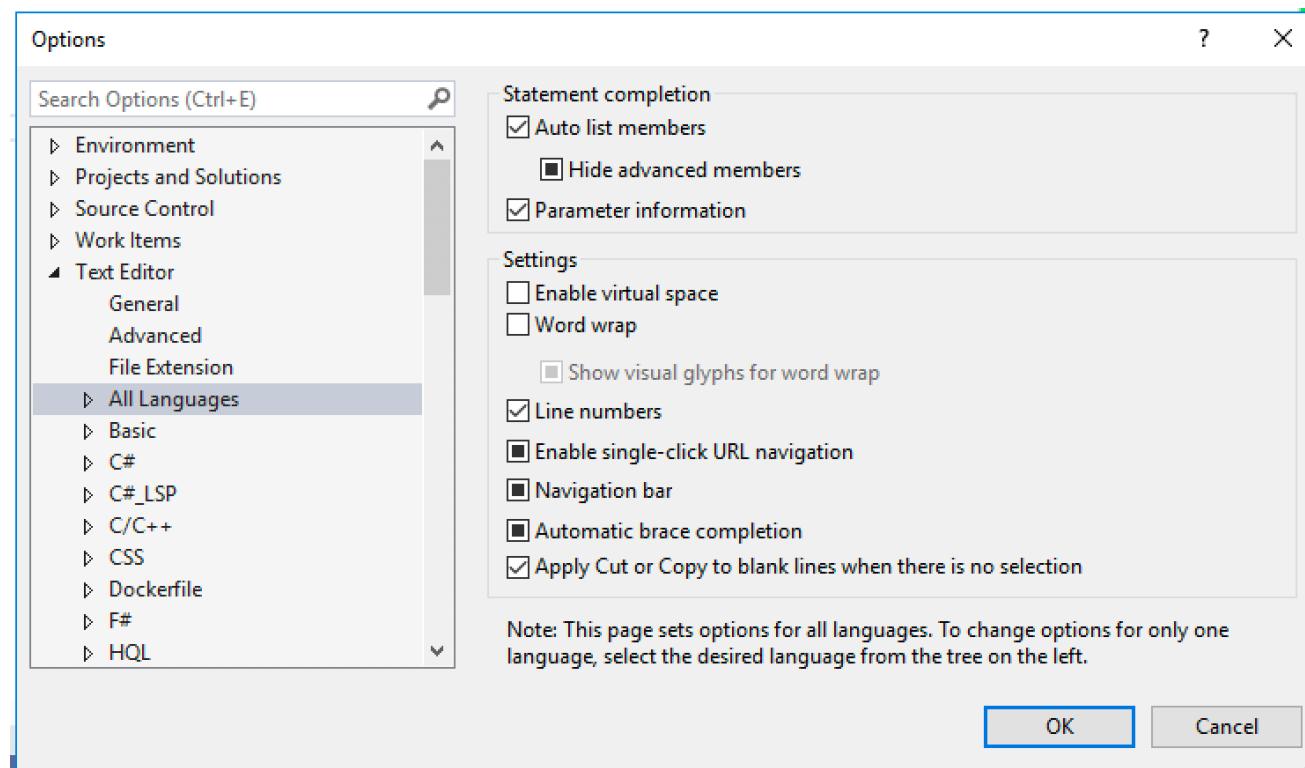
- Debug a sample program (**debug1.cpp**)
- Steps:
 - ▶ Create a new project in visual studio
 - ▶ Download the **debug1.cpp** from backboard and save it to the project folder
 - ▶ In visual studio, add an existing item to the project

Add the sample program to the project



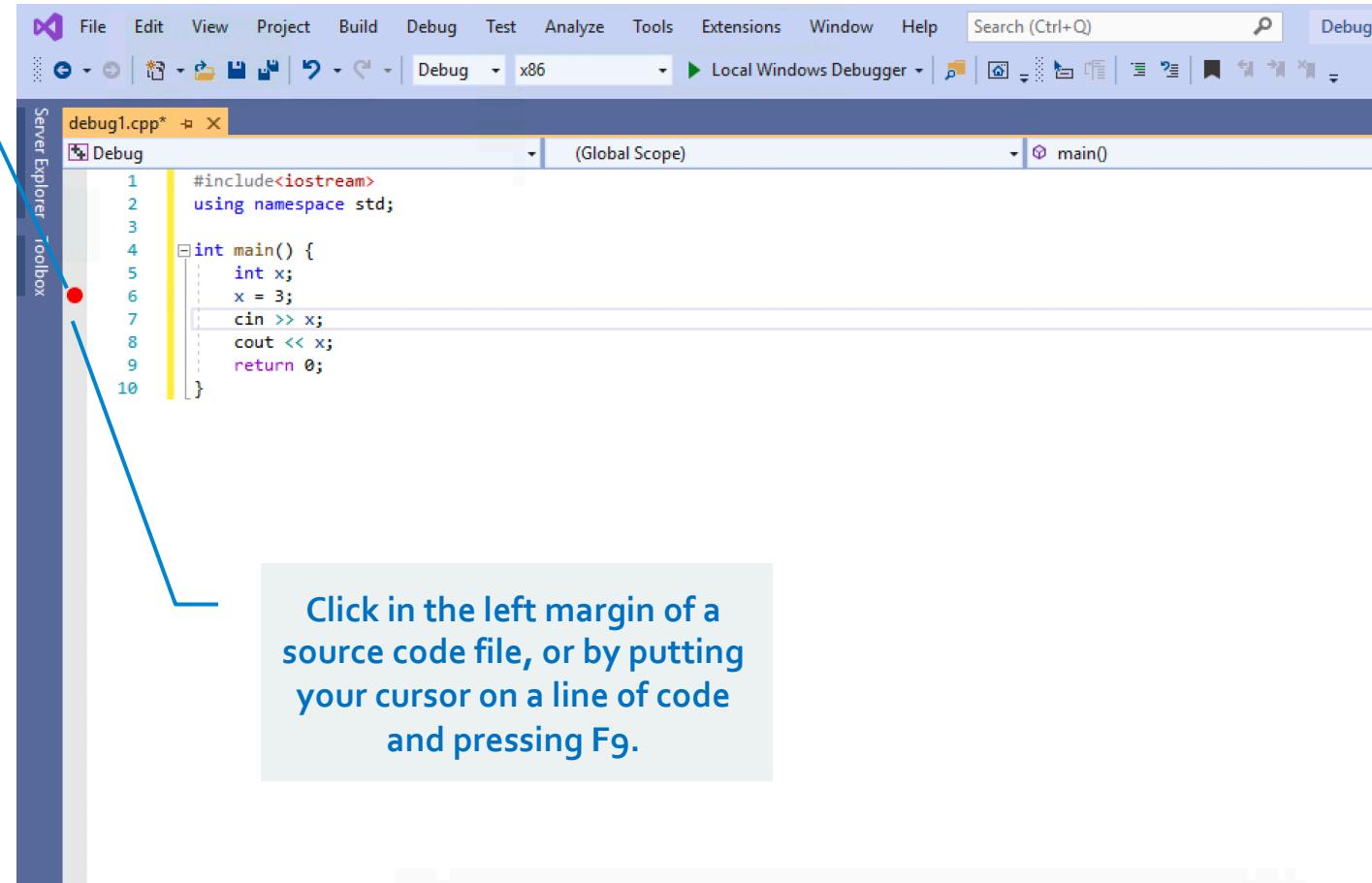
Tip: Display Line Numbers in Visual Studio

- 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.

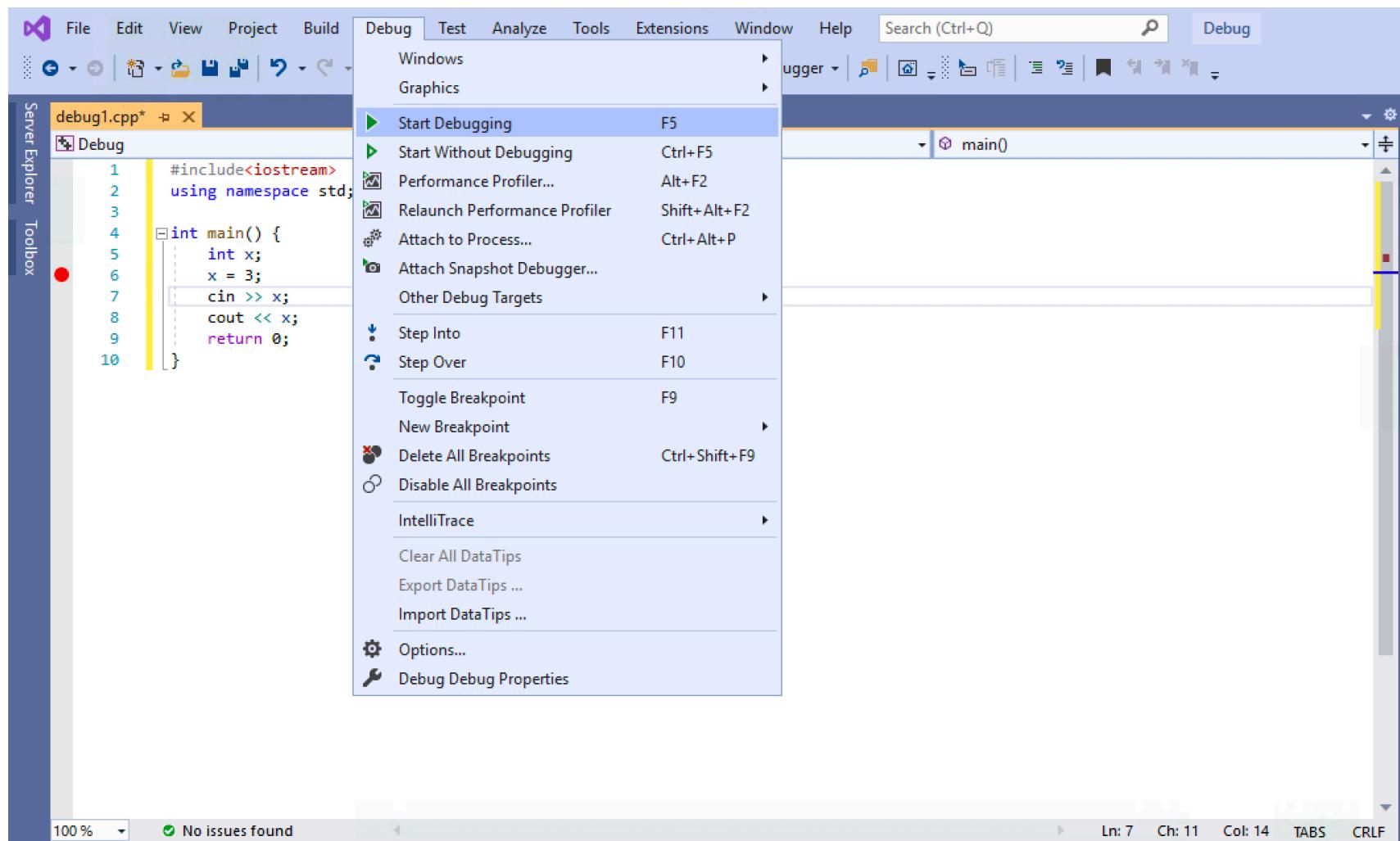


Click in the left margin of a source code file, or by putting your cursor on a line of code and pressing F9.

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**).  Local Windows Debugger ▾
- The program execution should stop at the line "**x=3**"

Start the debugger (menu bar)



Execution stop at x=3

The screenshot shows the Microsoft Visual Studio IDE interface during a debug session. The code editor displays `debug1.cpp` with the following content:

```
#include<iostream>
using namespace std;

int main() {
    int x;
    x = 3;
    cin >> x;
    cout << x;
    return 0;
}
```

A red circular breakpoint icon is positioned on the left margin of line 6. A blue arrow points from the text "Execution stops whenever the breakpoint is hit, before the code on that line is executed." to the breakpoint icon. The status bar at the bottom indicates "Ln: 6 Ch: 1 TABS CRLF".

The "Diagnostic Tools" window on the right shows performance metrics for the current process. The "Events" section is expanded, showing one event entry. The "Process Memory" section shows memory usage for the process. The "CPU (% of all processors)" section shows CPU usage across all processors.

The "Call Stack" window at the bottom shows the current call stack frame: `Debug.exe!main() Line 6 [External Code]`. The "Autos" window shows the variable `x` has a value of `-858993460` and is of type `int`.

Text overlay: Execution stops whenever the breakpoint is hit, before the code on that line is executed.

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.

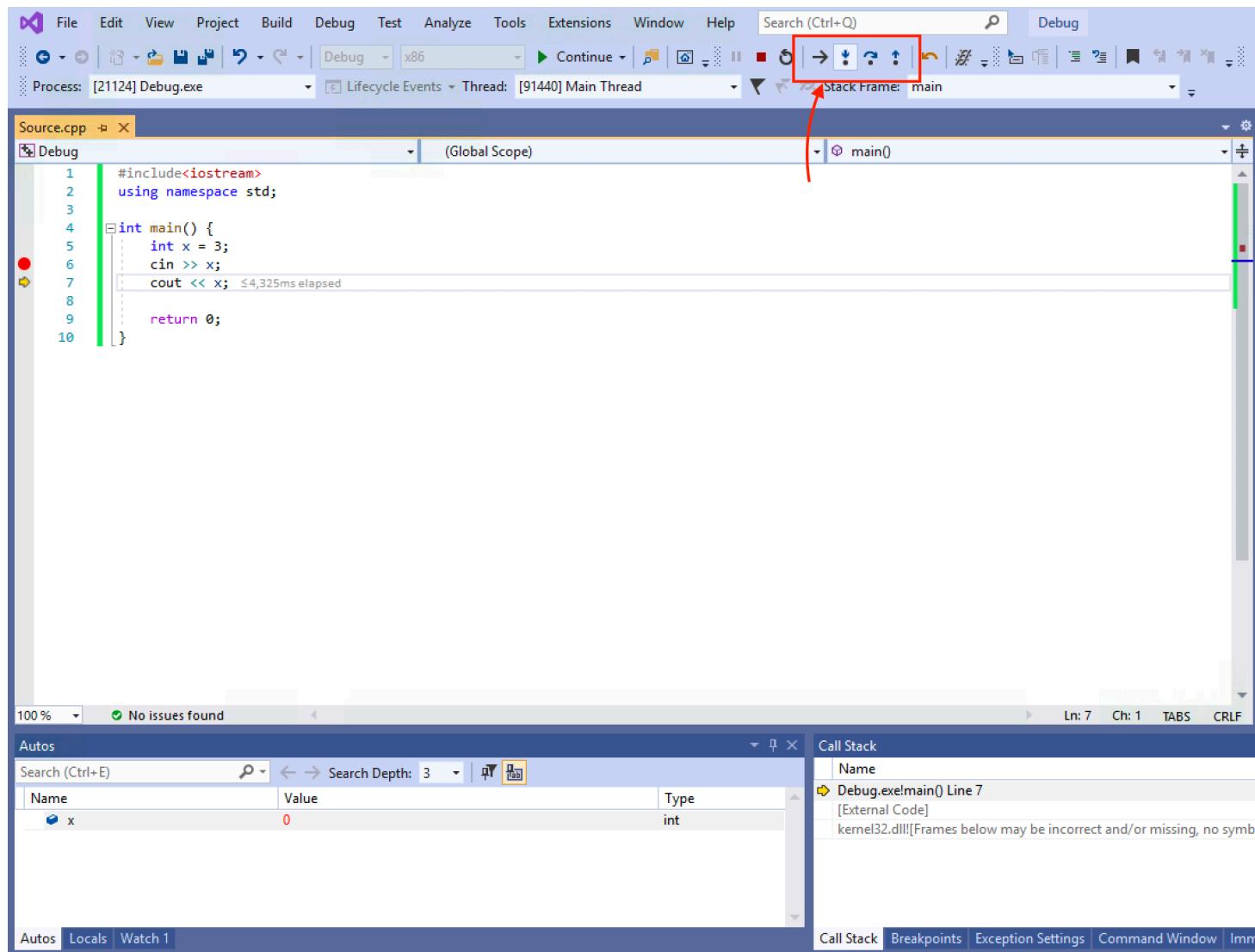
- **Step Into (F11):**

- ▶ 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



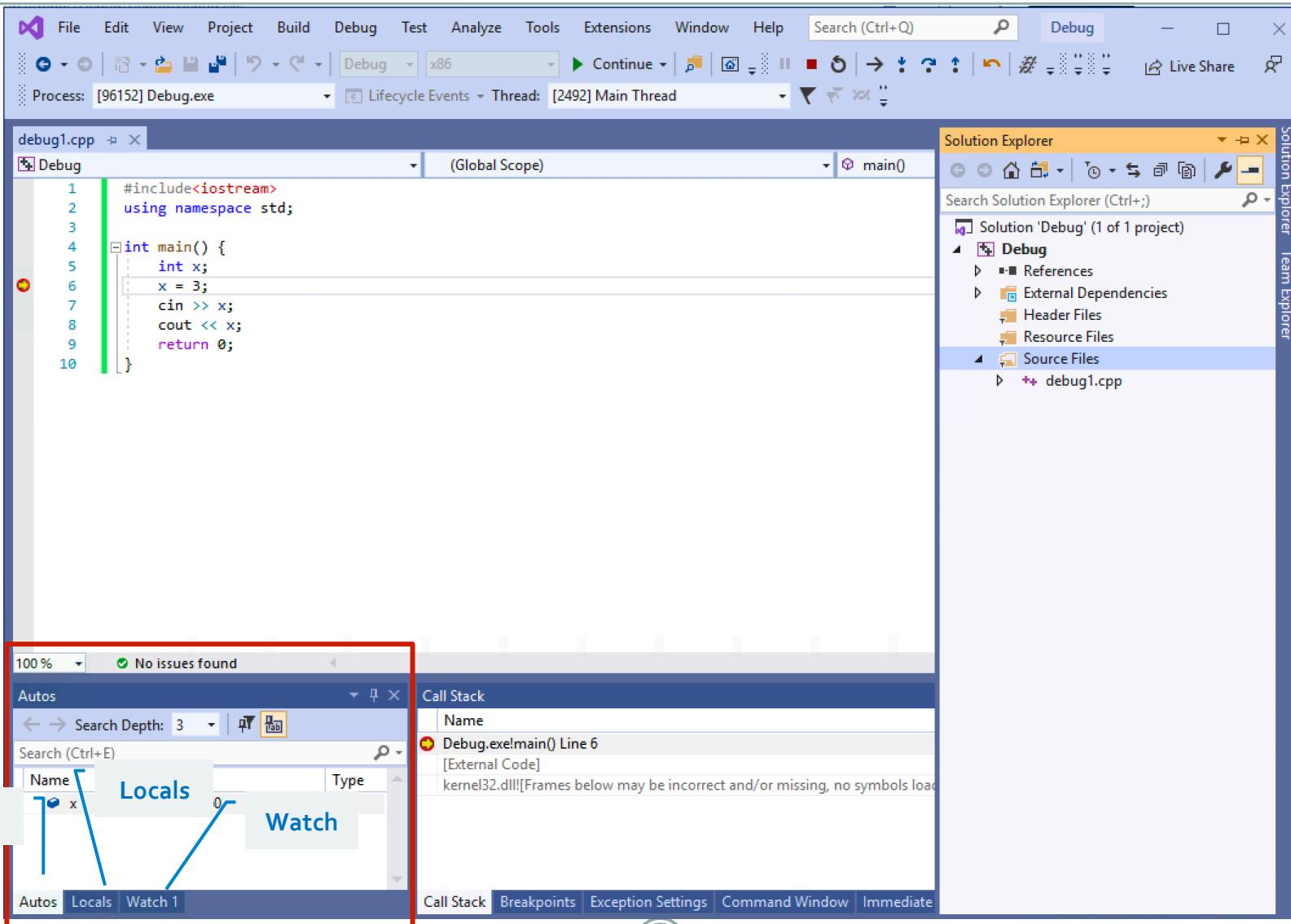
Summary of short-cut key

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
 - ▶ **Watch1..N**: display user selected variable

Display the value of a variable



Example: $x = -858993460$

The screenshot shows a Microsoft Visual Studio interface during a debugging session. The code editor displays `debug1.cpp` with the following content:

```
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 }
```

A red circle highlights the breakpoint at line 6. The status bar indicates "100%" and "No issues found".

The Solution Explorer window shows a single project "Debug" with files: References, External Dependencies, Header Files, Resource Files, Source Files (containing `debug1.cpp`).

The Call Stack window shows the current call stack entry:

Name
Debug.exe!main() Line 6 [External Code]

The Autos window displays the variable `x` with a value of `-858993460`, which is highlighted with a red border.

Example: $x=3$

debug1.cpp

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

Process: [96152] Debug.exe Lifecycle Events - Thread: [2492] Main Thread

Debug (Global Scope) main()

Call Stack

Name	Type
Debug.exe!main() Line 7	[External Code]

Autos

Name	Value	Type
x	3	int

Call Stack Breakpoints Exception Settings Command Window Immediate Window Output

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

The screenshot shows a Microsoft Visual Studio interface during a debugging session. The top menu bar includes File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a Search bar. The Debug tab is selected. The toolbar below has various icons for debugging, including a red stop button, a green start button, and a yellow resume button. The status bar at the bottom shows "Process: [96152] Debug.exe".

The code editor window displays "debug1.cpp" with the following content:

```
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 }
```

A red dot on the left margin indicates a breakpoint at line 4. A tooltip above the cursor shows the current value of variable `x` as `5`. The immediate window shows the command `C:\Users\xinlinli\source\repos\Debug\Debug\Debug.exe`.

The bottom pane contains the Autos, Locals, and Watch 1 tabs. The Autos tab shows a table with one row:

Name	Value
x	3

The Locals and Watch 1 tabs are currently inactive.

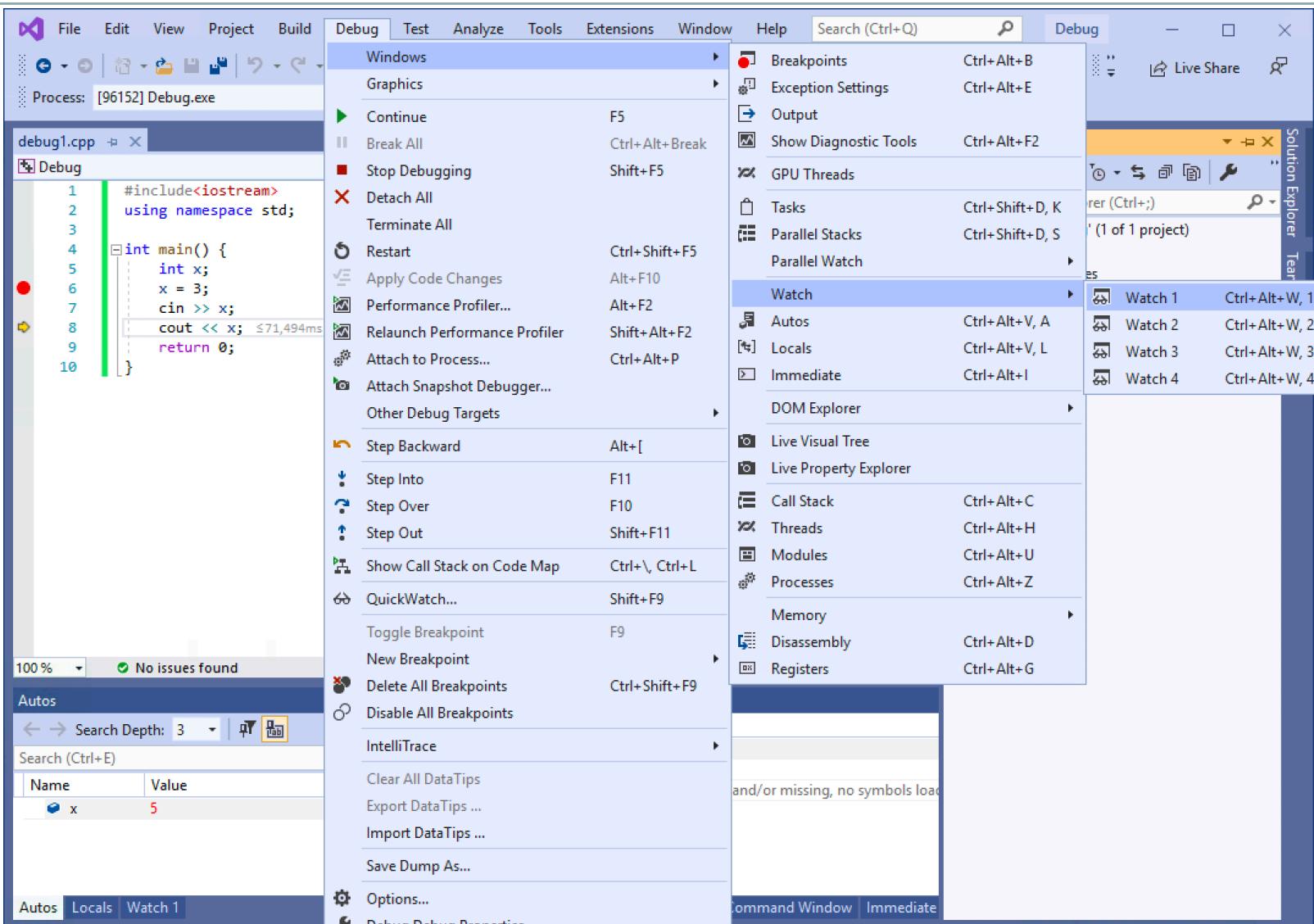
Example: $x=5$

The screenshot shows the Microsoft Visual Studio IDE interface during a debugging session. The main window displays the code for `debug1.cpp` in the Global Scope. A red dot at line 8 indicates a breakpoint. The code reads:

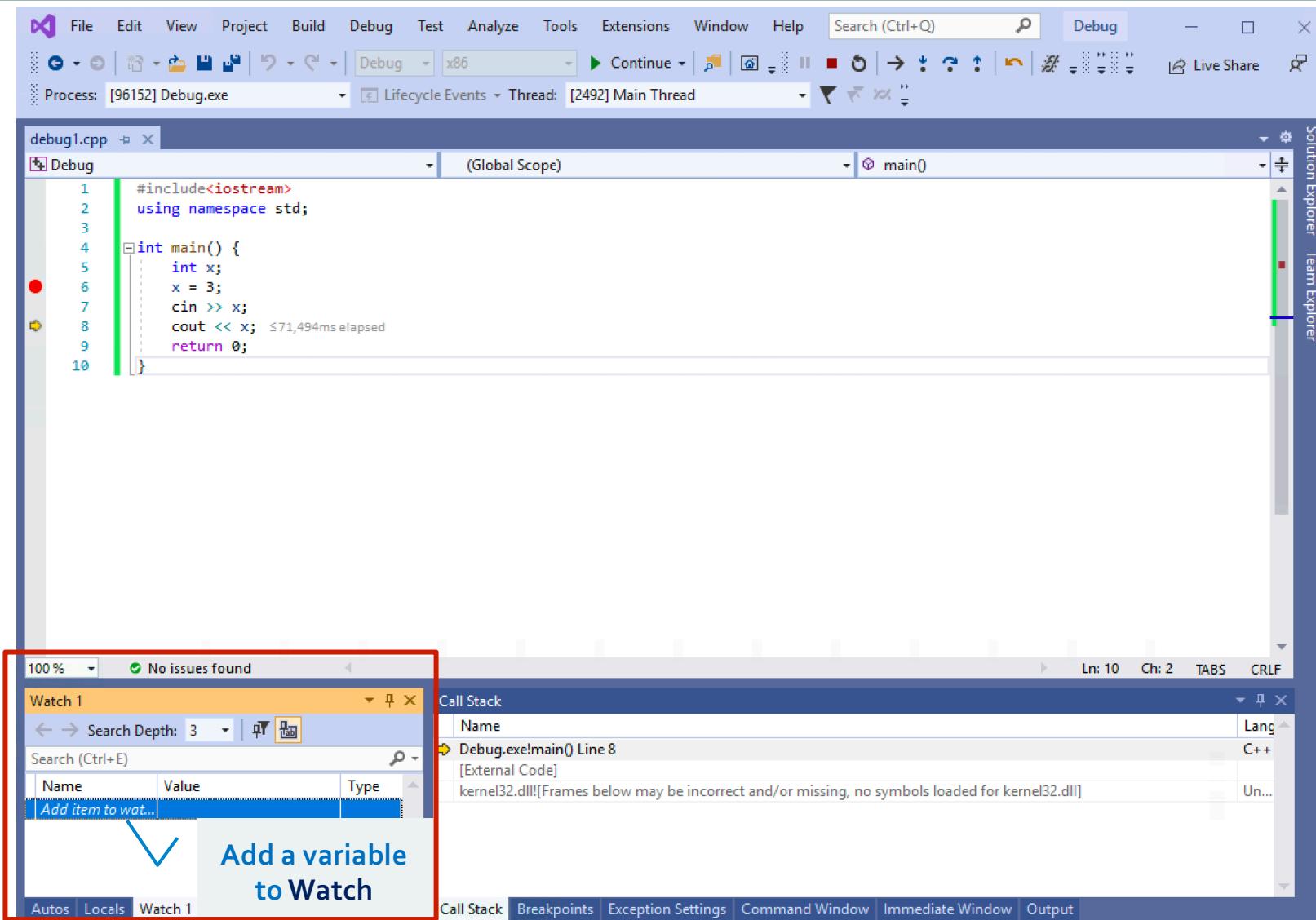
```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     int x;
6     x = 3;
7     cin >> x;
8     cout << x; // 71,494ms elapsed
9     return 0;
10 }
```

The Autos window shows the variable `x` has a value of `5`. The Call Stack window shows the current call stack entry is `Debug.exe!main() Line 8 [External Code]`. The Solution Explorer window shows the project structure with `debug1.cpp` selected.

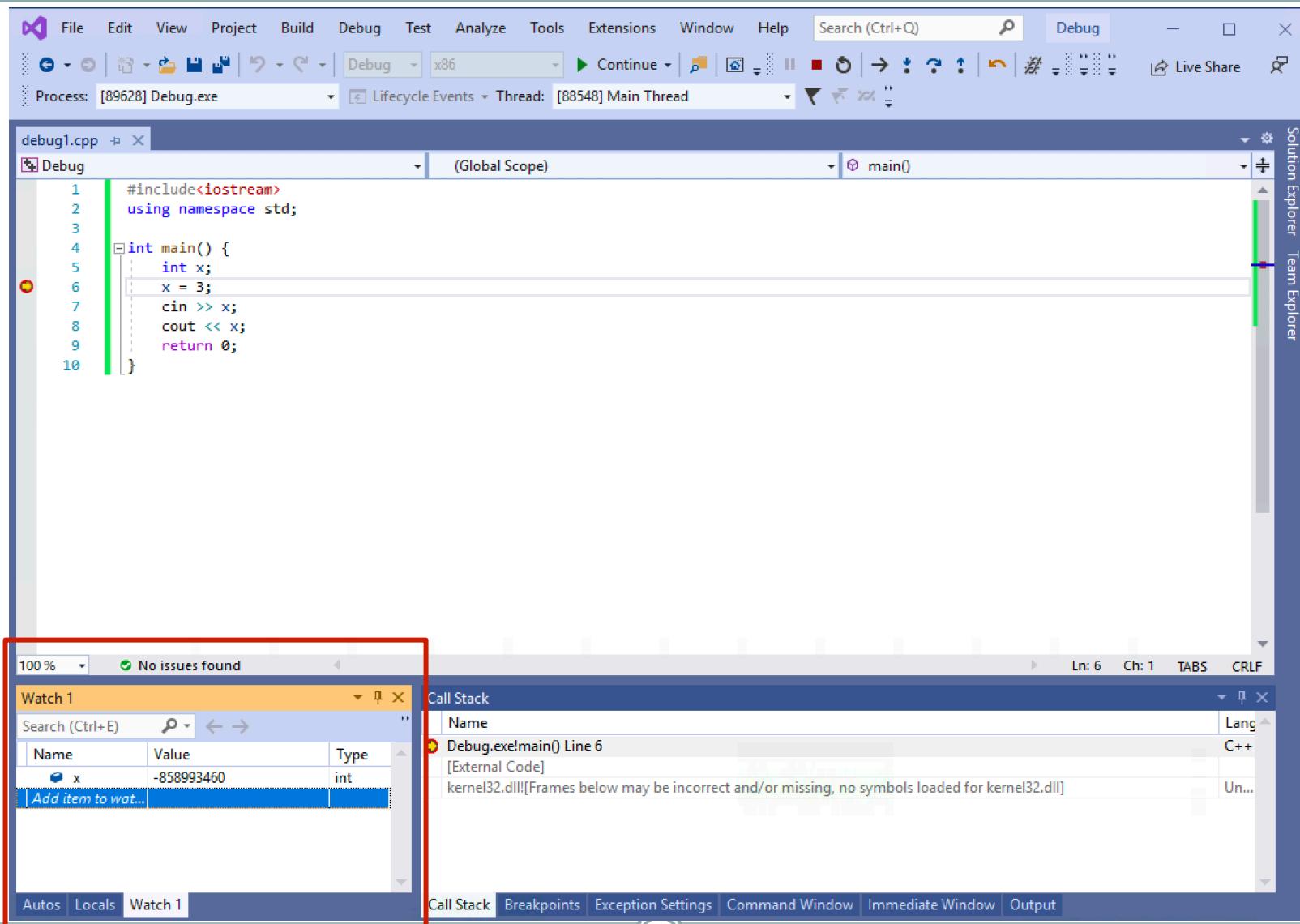
If you can't find the Watch windows, ...



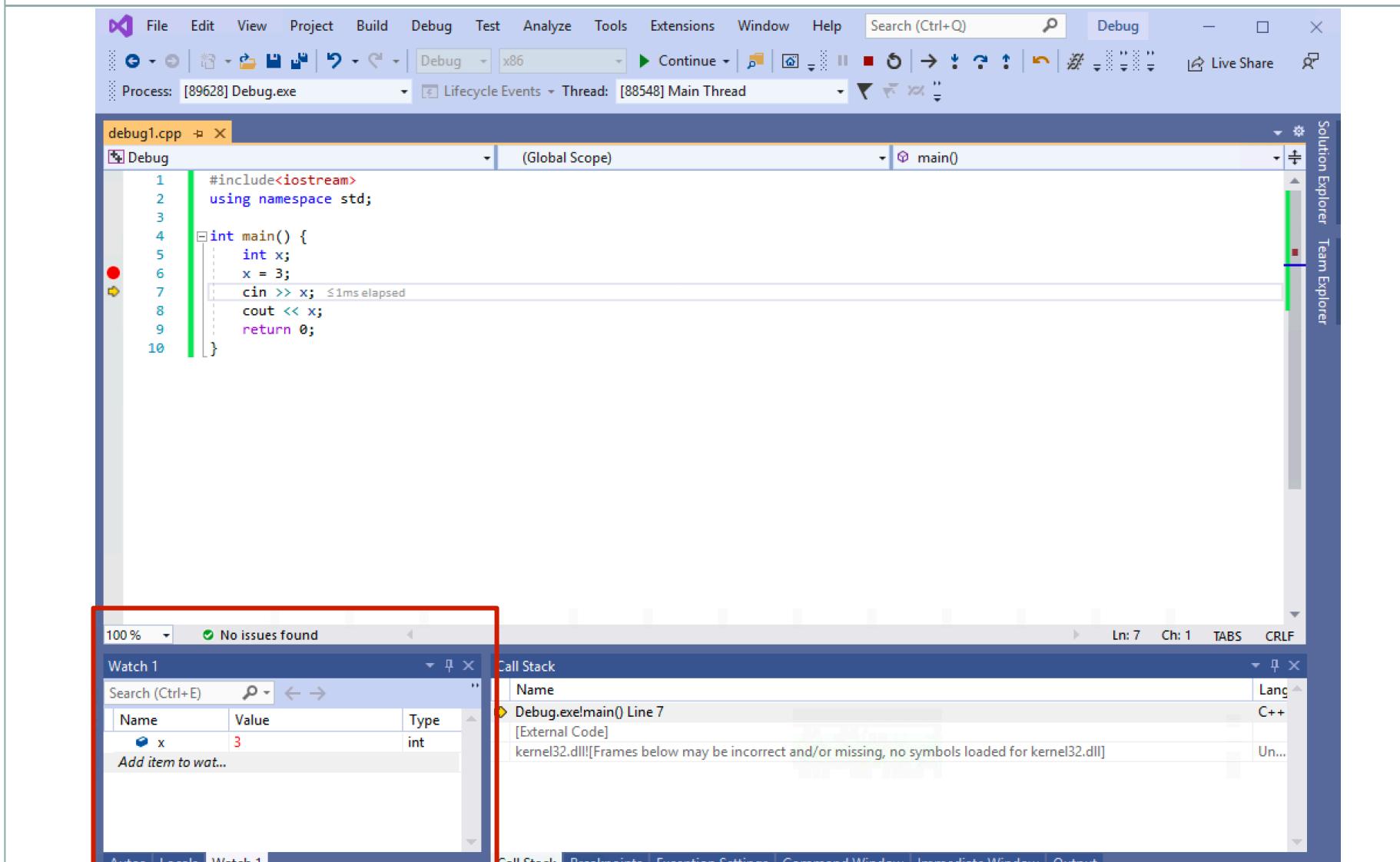
Add a variable to Watch



Add a variable to Watch (cont'd)

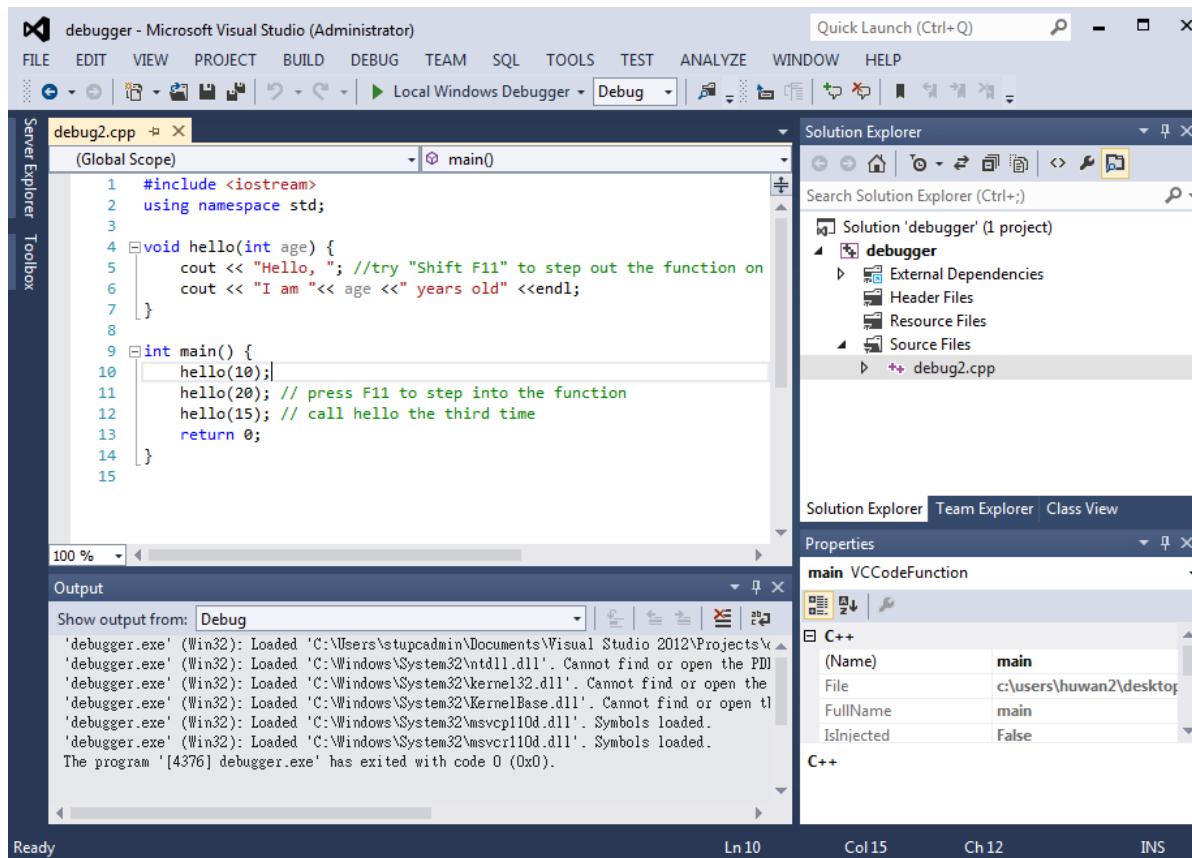


Add a variable to Watch (cont'd)



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

The screenshot shows a Microsoft Visual Studio interface with the title bar "debugger - Microsoft Visual Studio (Administrator)". The menu bar includes FILE, EDIT, VIEW, PROJECT, BUILD, DEBUG, TEAM, SQL, TOOLS, TEST, ANALYZE, WINDOW, and HELP. The toolbar has various icons for file operations. A status bar at the bottom says "Ready".

The code editor displays two files: "cToF.cpp" and "crtexe.c". The "cToF.cpp" file contains the following C++ code:

```
5 #include<iostream>
6 using namespace std;
7
8 int main(){
9     double C = 0;
10    double F = 0;
11    double factor;
12
13    cout << "Input temperature in Fahrenheit: ";
14    cin >> F;
15    factor = 5 / 9;
16    C = (F - 32) * factor;
17    cout << "Temperature in Celius: " << C;
18    return 0;
19 }
20
```

The output window shows the command prompt "C:\Windows\system32\cmd.exe" and the execution of the program:

```
Input temperature in Fahrenheit:100
Temperature in Celius: 0
Press any key to continue . . .
```

Exercise 3

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

The screenshot shows the Microsoft Visual Studio IDE interface during debugging. The main window displays the code for `nestedfor.cpp` in the editor. A red dot at line 10 indicates the current execution point. The output window shows the program's execution:

```
C:\Windows\system32\cmd.exe
6
65
654
6543
65432
654321
Press any key to continue . . .
```

Exercise 4

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

The screenshot shows the Microsoft Visual Studio interface with the following windows open:

- Code Editor:** Displays the file `sumArray.cpp` with the following code:

```
1  /*debug the following program which sum all the elements for array n*/
2
3  #include <iostream>
4  using namespace std;
5
6 int main(){
7     int n[3] = {5, 10, 5};
8     int sum, i = 0;
9     while(i <= 3){
10         i++;
11         sum += n[i];
12     }
13     cout << sum << endl;
14     return 0;
15 }
```
- Solution Explorer:** Shows the project structure for "debugger".
 - External Dependencies
 - Header Files
 - Resource Files
 - Source Files
 - sumArray.cpp
- Error List:** Shows 1 error:

Description	File	Line	Column	Project
error C4700: uninitialized local variable 'sum' used	sumarray.cpp	11	1	debugger
- Properties:** Shows the properties for the `main` function.

C++	(Name)	main
File	File	c:\users\huwan2\desktop\
FullName	FullName	main
IsInjected	IsInjected	False