

CS2311 Computer Programming

Labo6: Debug a Program with Visual Studio

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

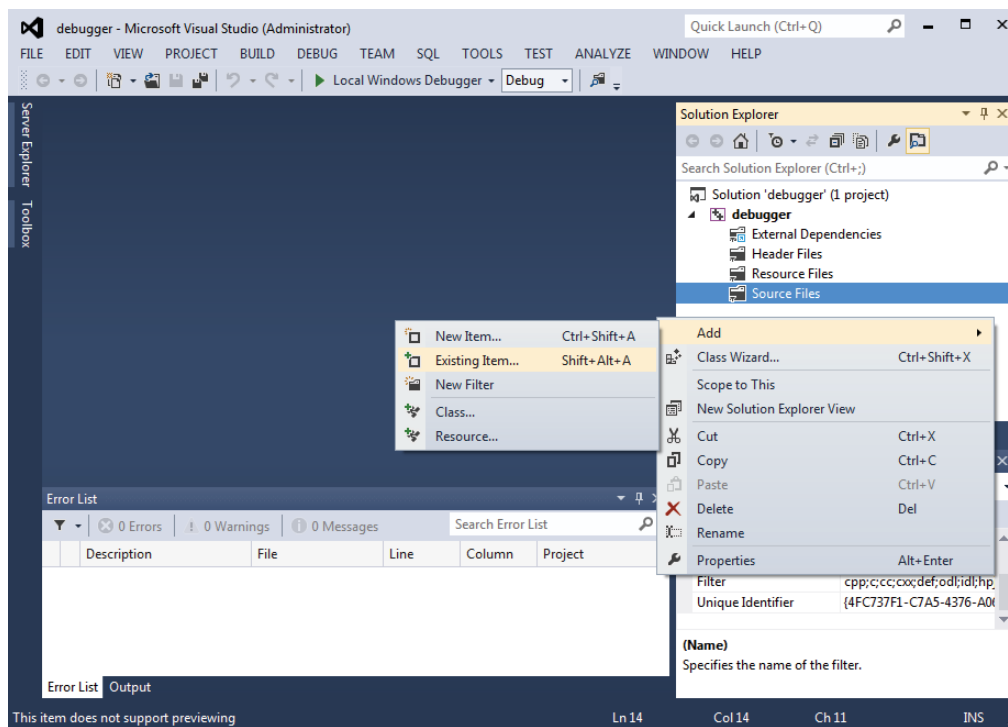
Outlines

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

Lab Exercise

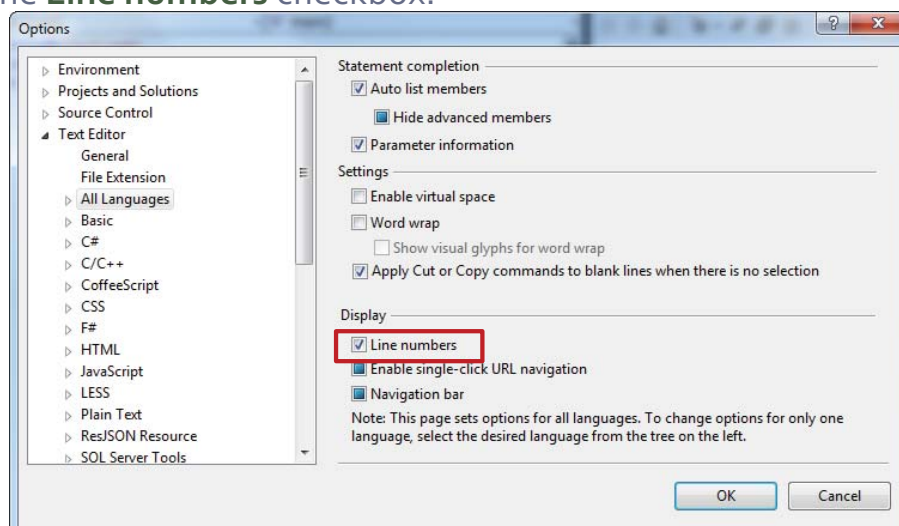
- Debug a sample program (**debug1.cpp**)
- Steps:
 1. Create a new project in visual studio
 2. Download the **debug1.cpp** from backboard and save it to the project folder
 3. In visual studio, add an existing item to the project

Add the sample program to the project



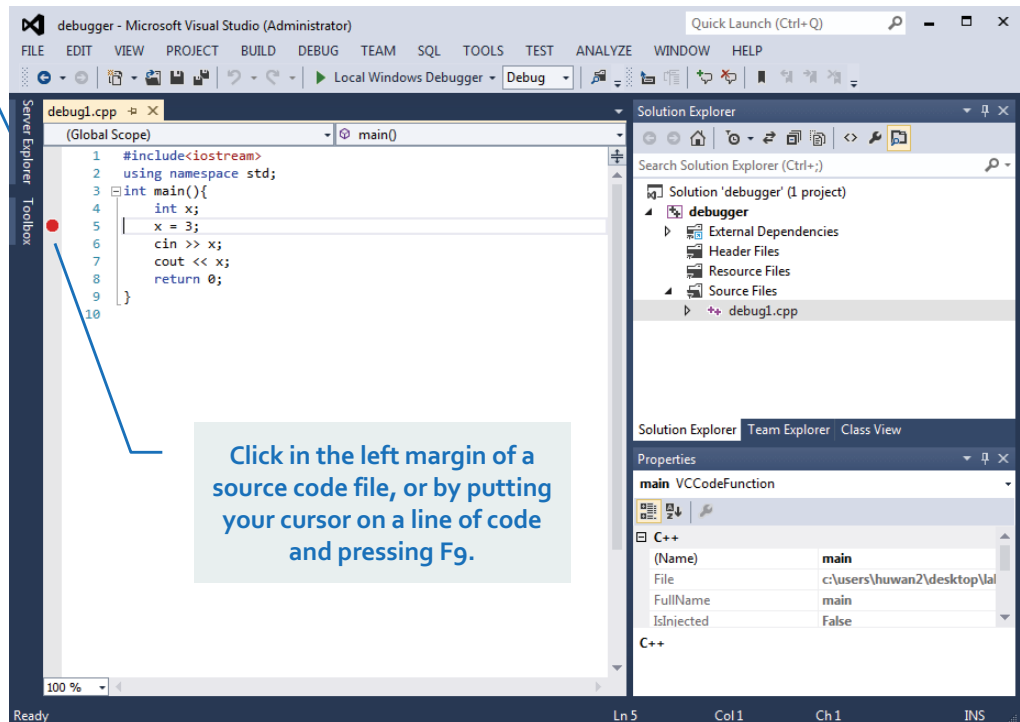
Tip: Display Line Numbers in Visual Studio

- 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.

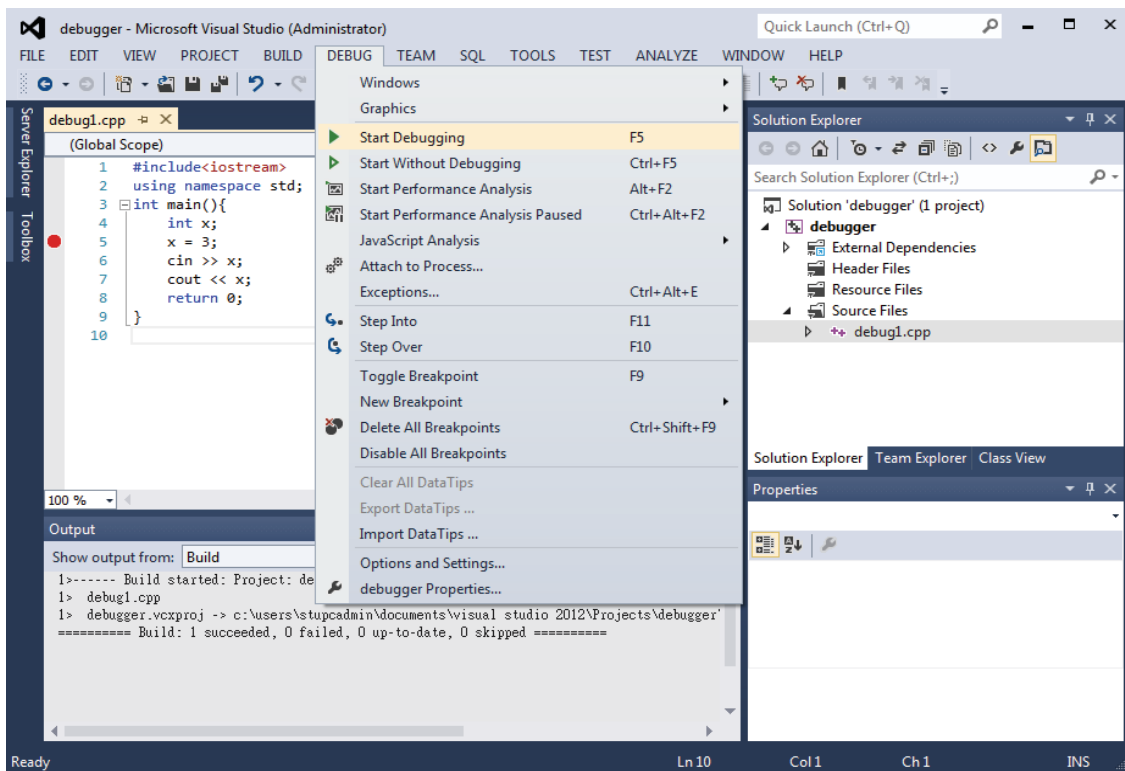


Click in the left margin of a source code file, or by putting your cursor on a line of code and pressing F9.

Steps to invoke the debugger

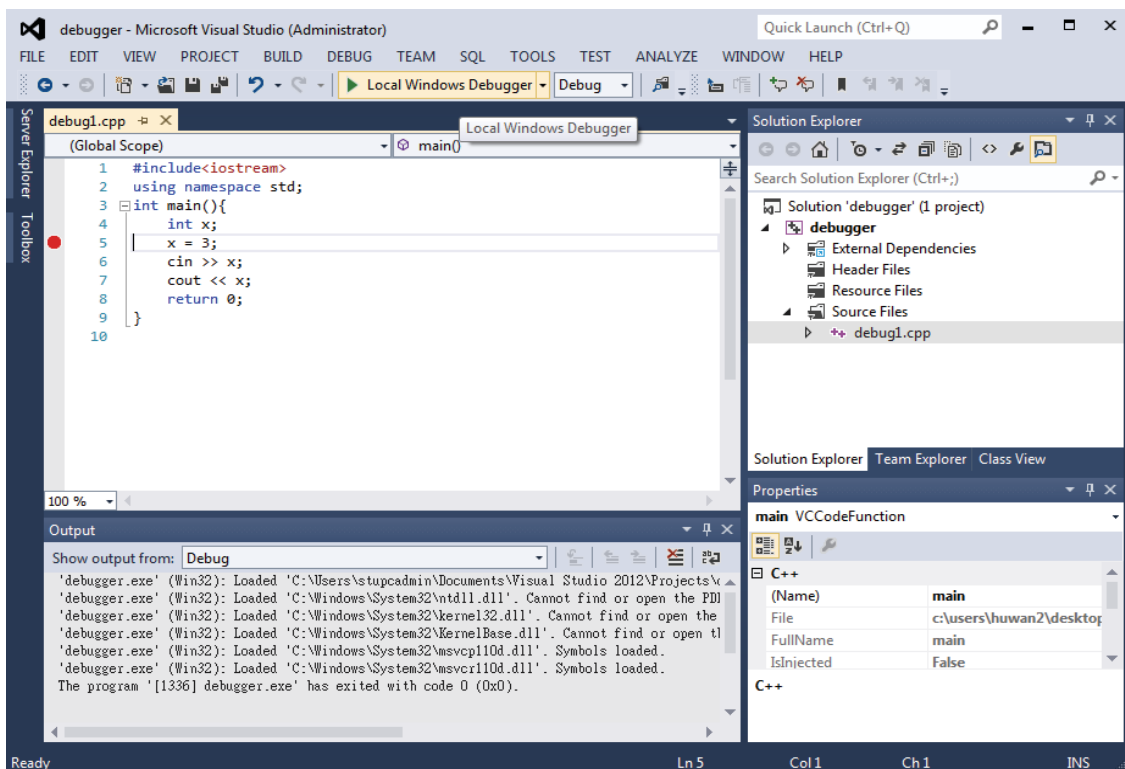
- Build/Rebuild the solution
- Start the debugger
 - ▶ Start a debugging session using **F5** (**Debug > Start Debugging**). This command starts your app with the debugger attached.
 - ▶ The green arrow on the tool bar also starts the debugger (same as **F5**).
- The program execution should stop at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

Start the debugger (menu bar)



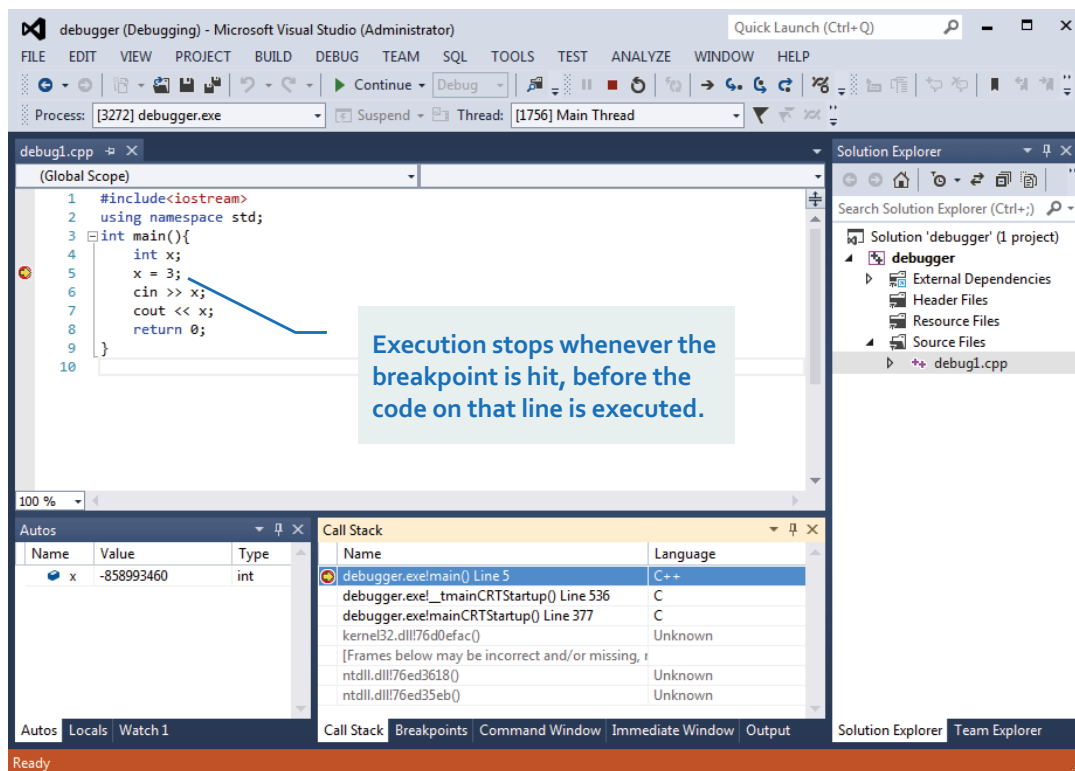
9

Start the debugger (tool bar)



10

Execution stop at x=3



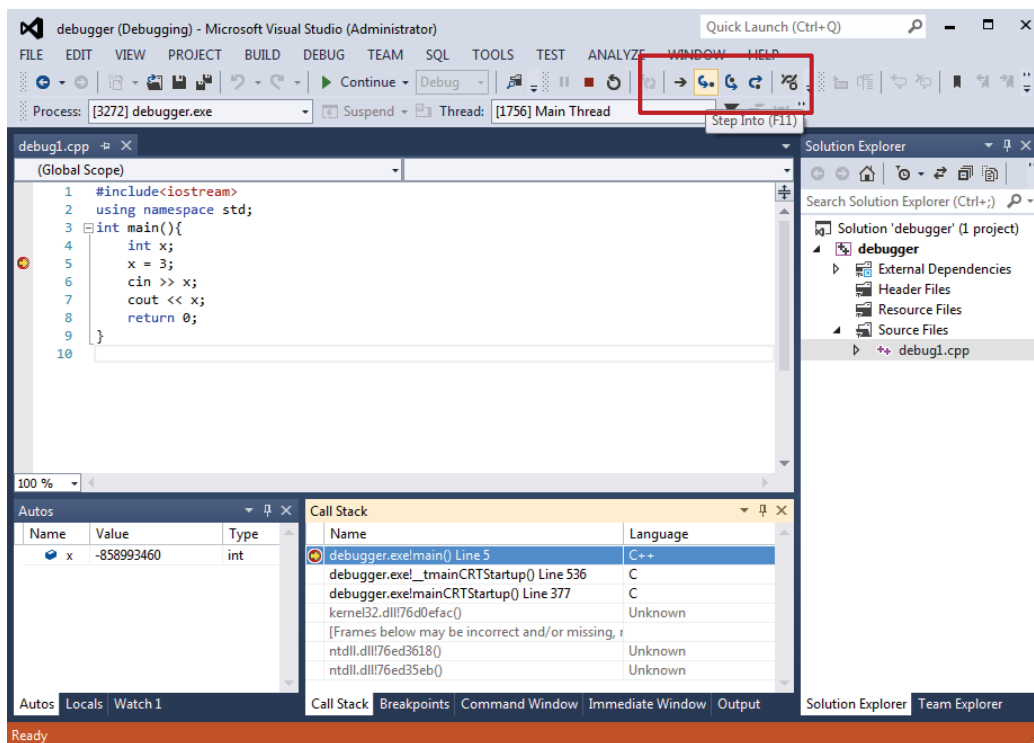
11

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 called.
- **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.

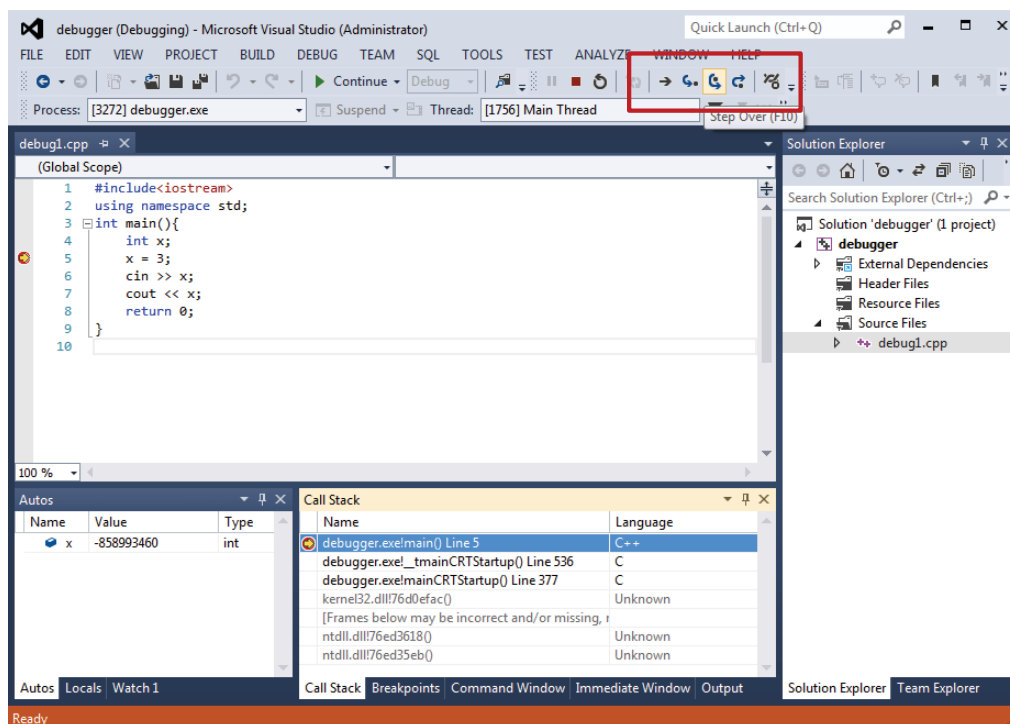
12

Step into code, line by line



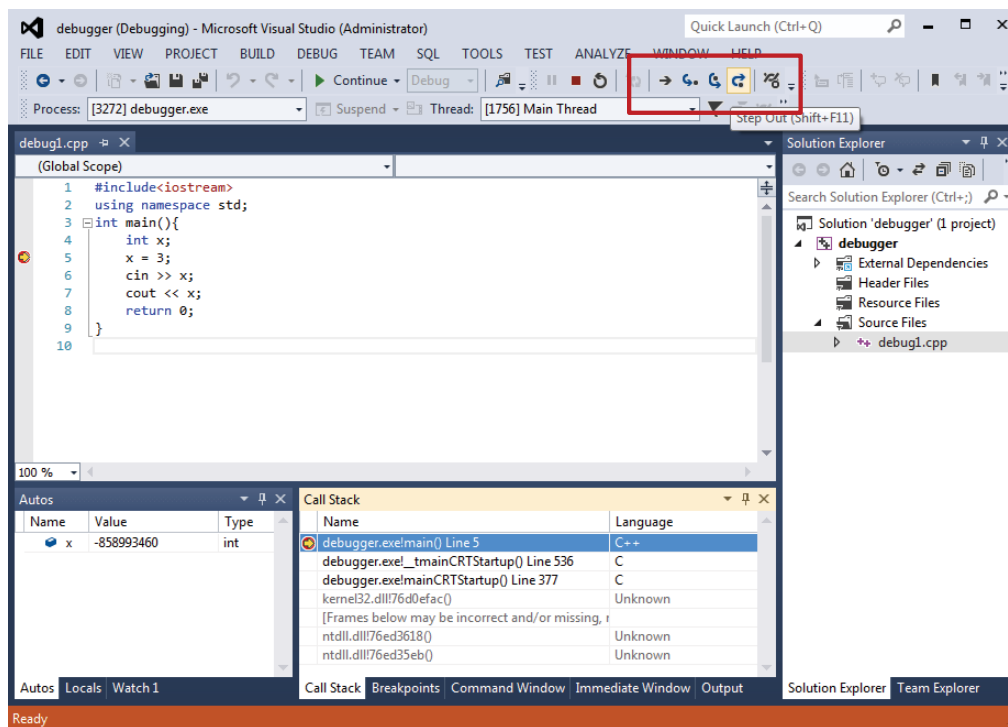
13

Step through code, skipping functions



14

Step through code, skipping functions



15

Summary of *short-cut key*

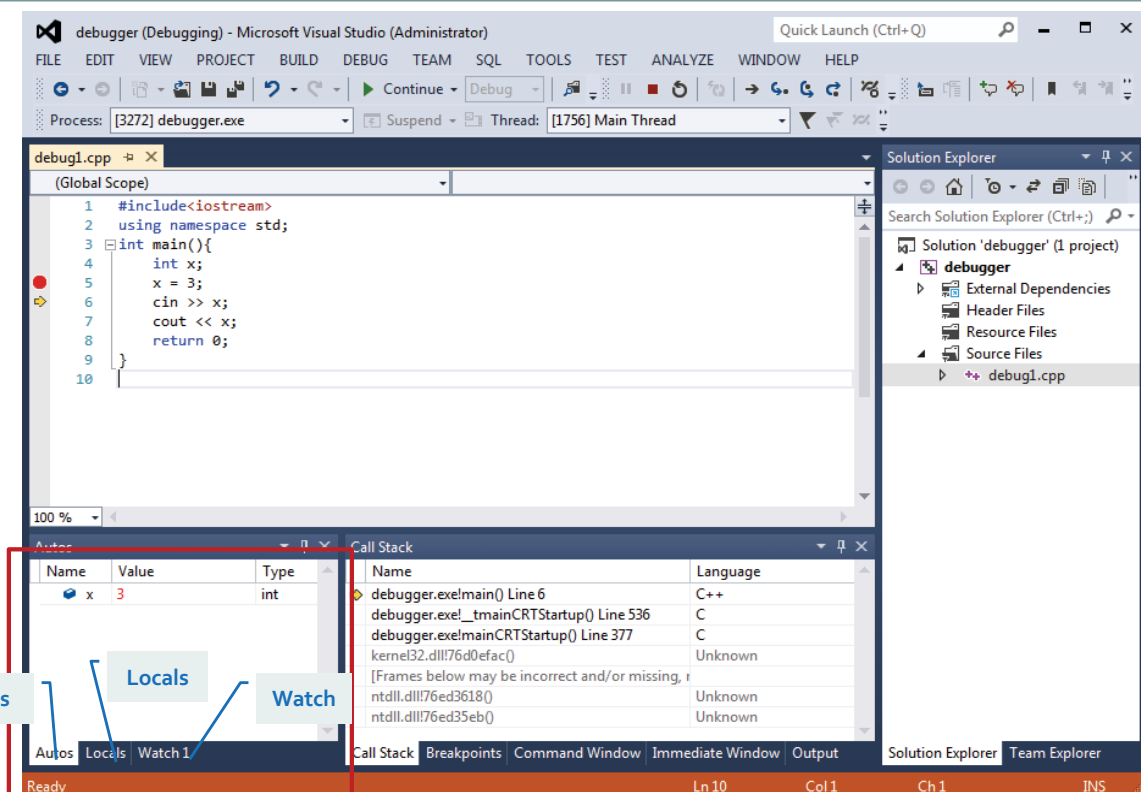
Key	Function
Ctrl+alt+F7	Rebuild the solution
F5	Start debugging
Ctrl+F5	Start without debugging
Shift+F5	Stop debugging
F10	Step over
F11	Step into
Shift + F11	Step out

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

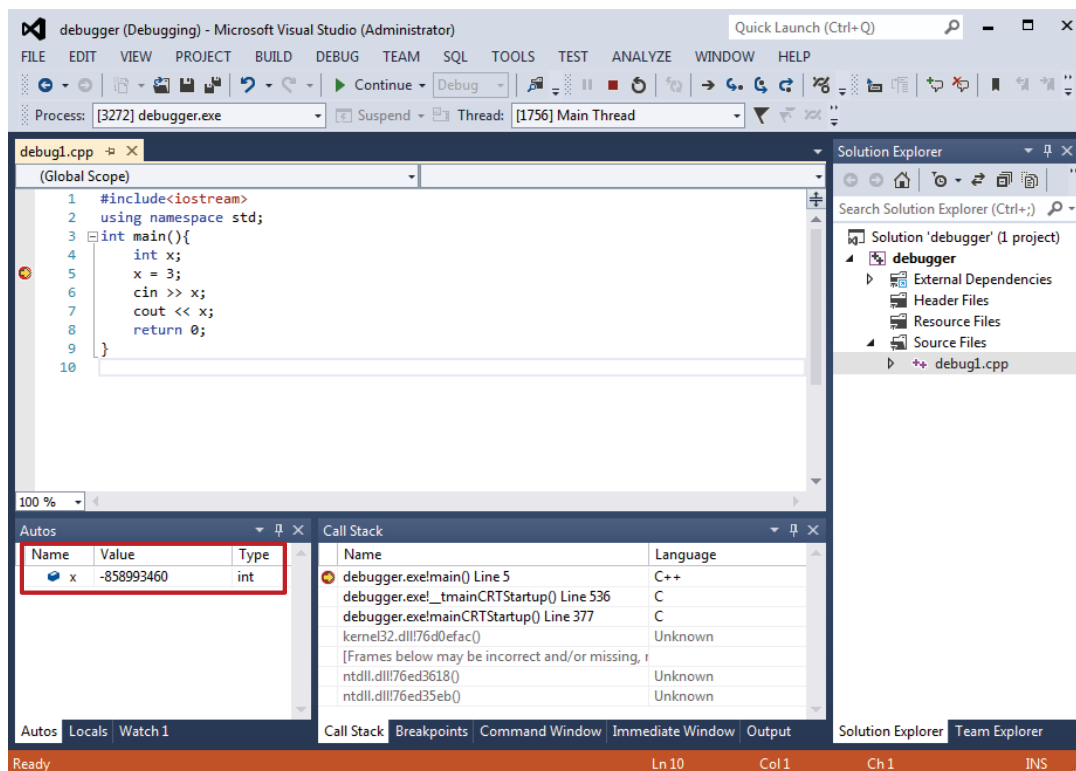
17

Display the value of a variable



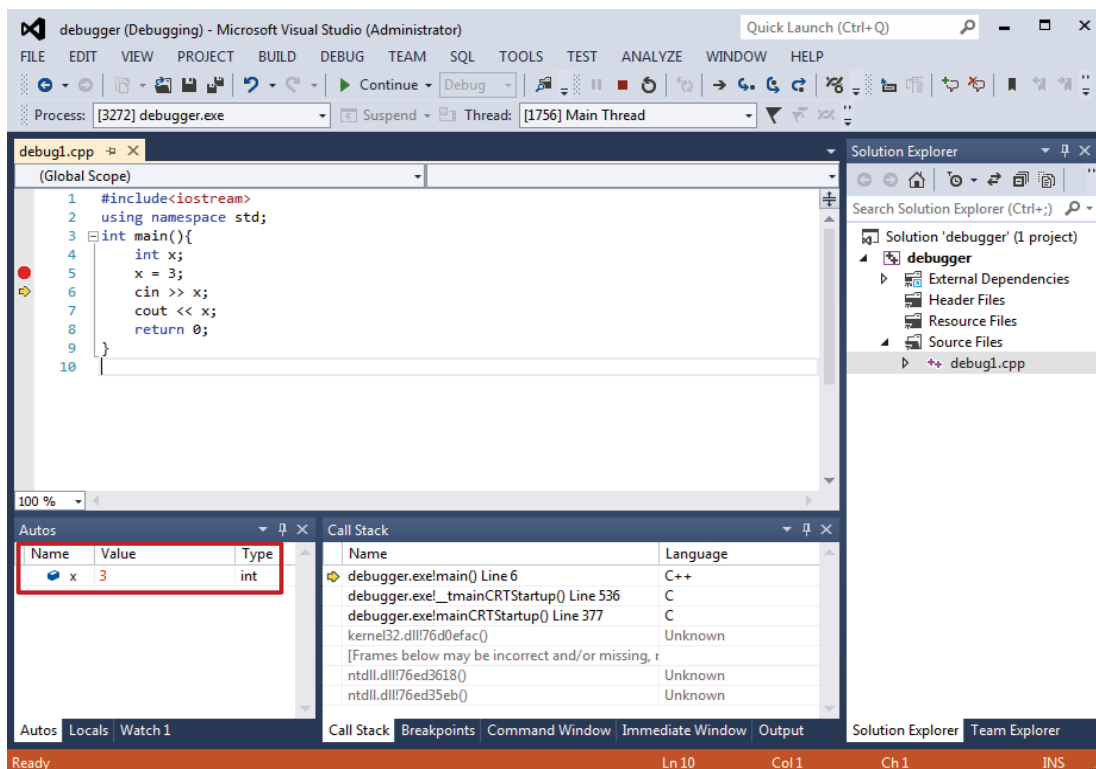
18

Example: x = -858993460



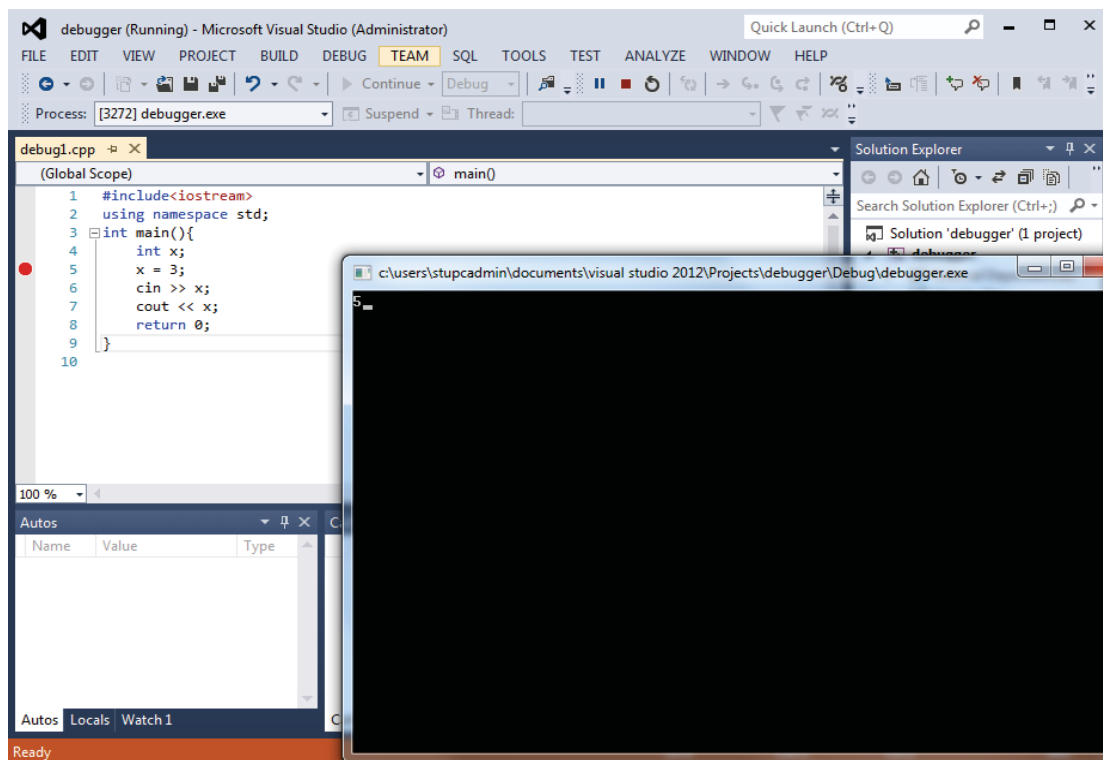
19

Example: x = 3



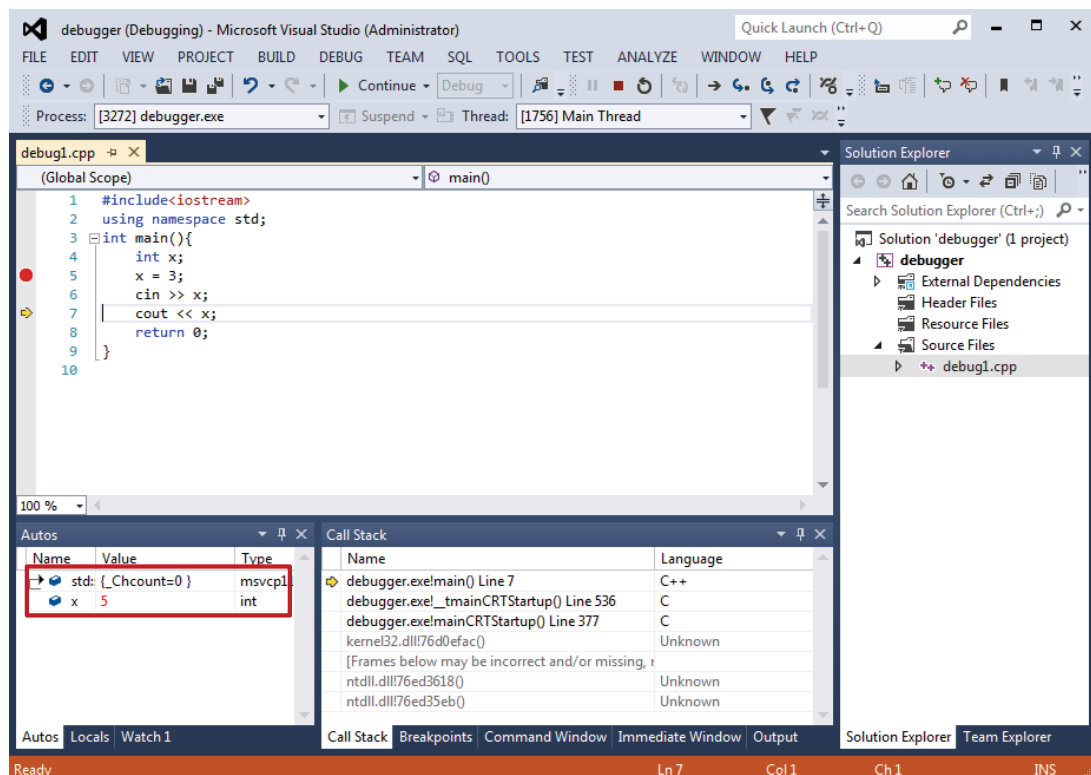
20

Example: input new x



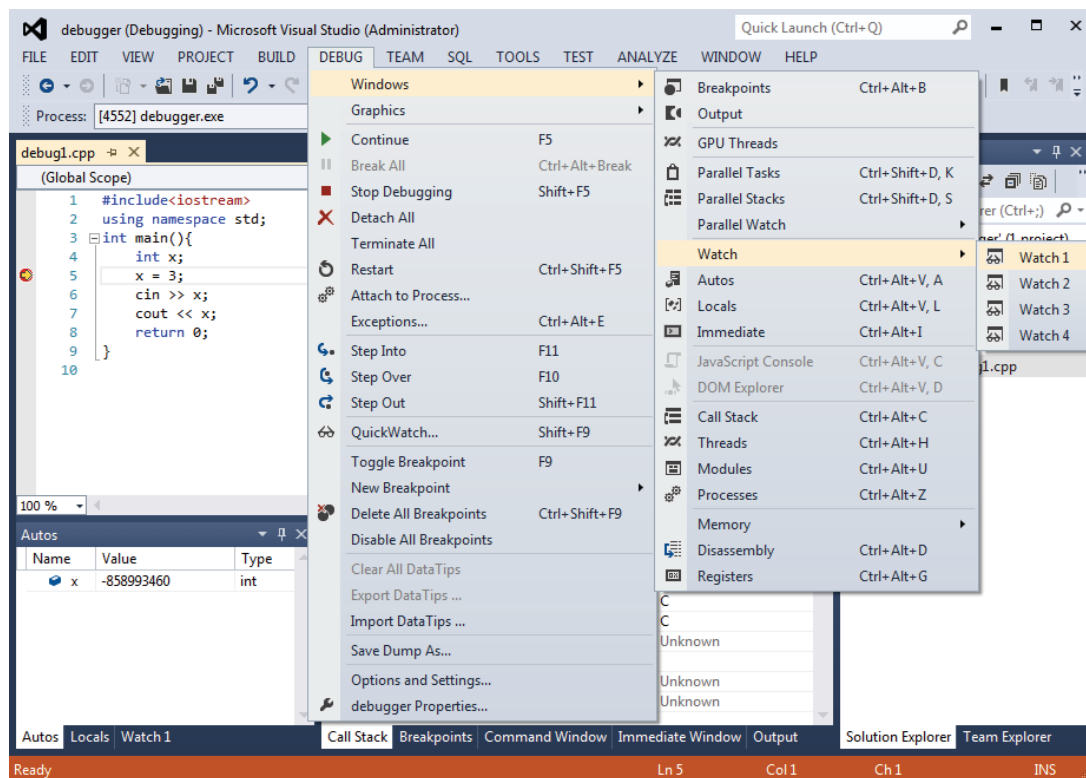
21

Example: x = 5



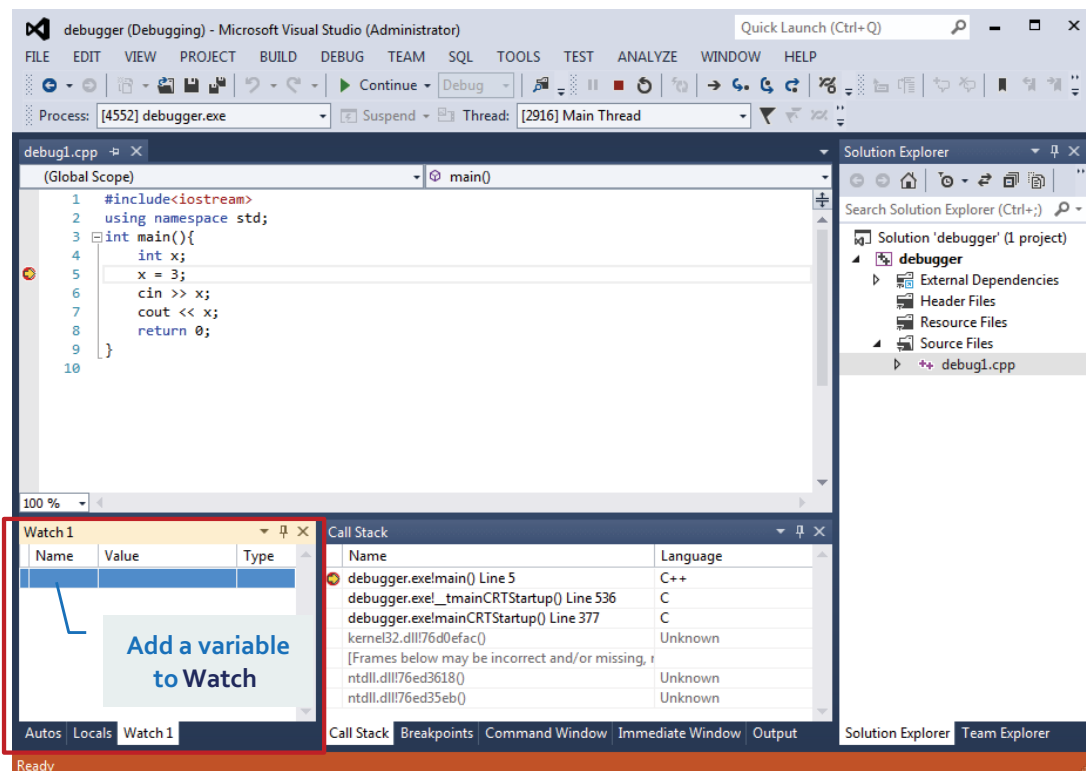
22

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



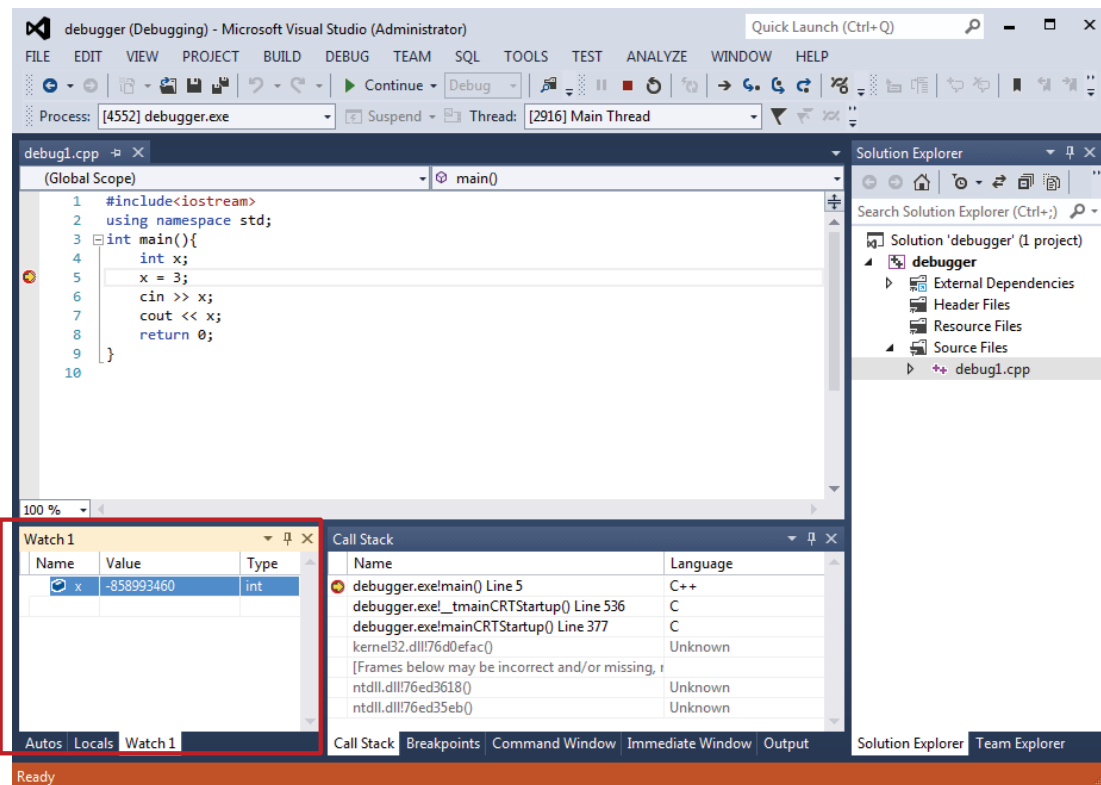
23

Add a variable to Watch



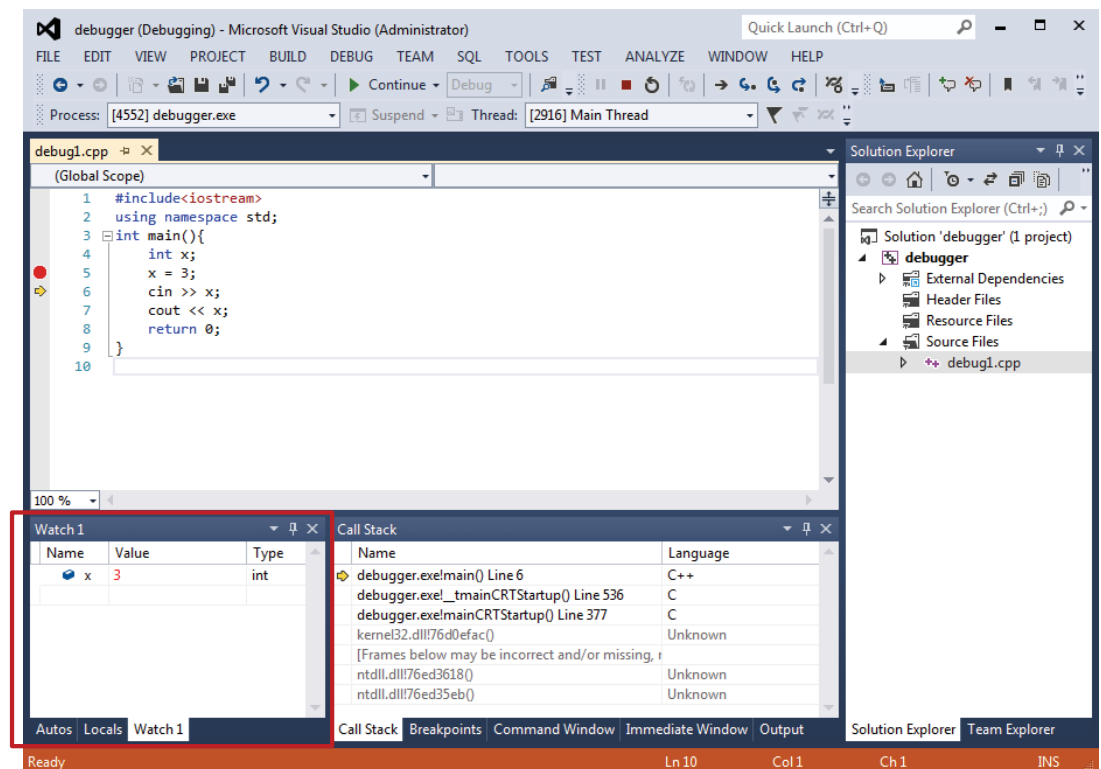
24

Add a variable to Watch (cont'd)



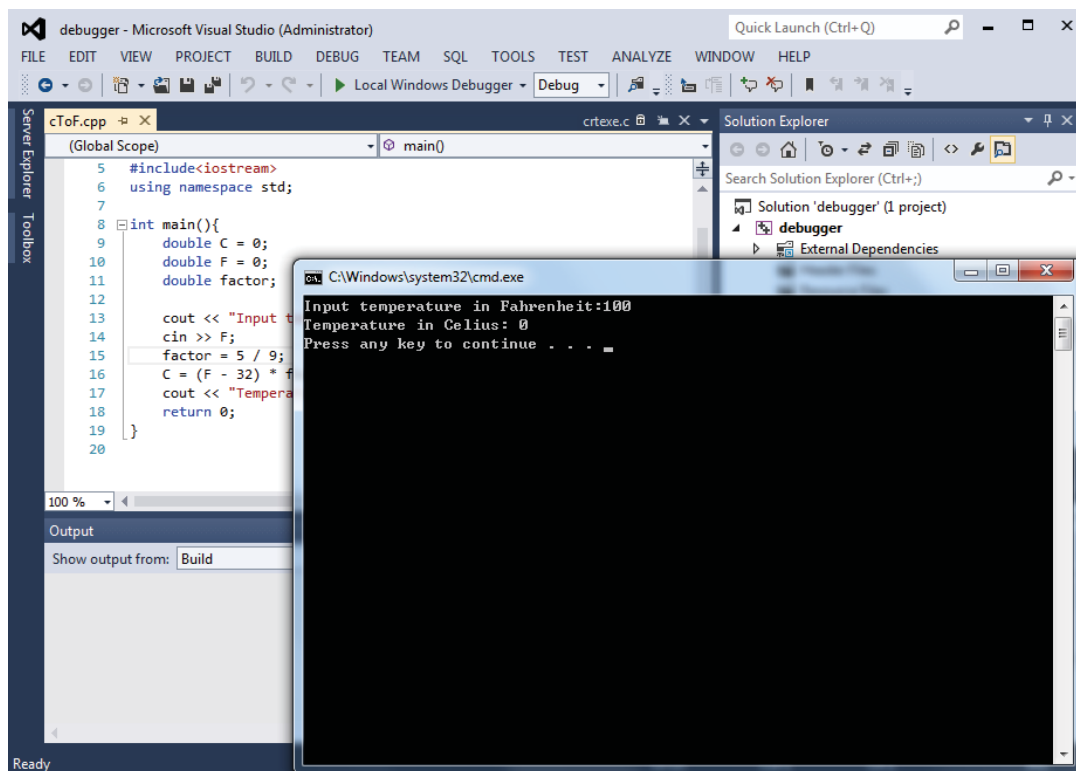
25

Add a variable to Watch (cont'd)



26

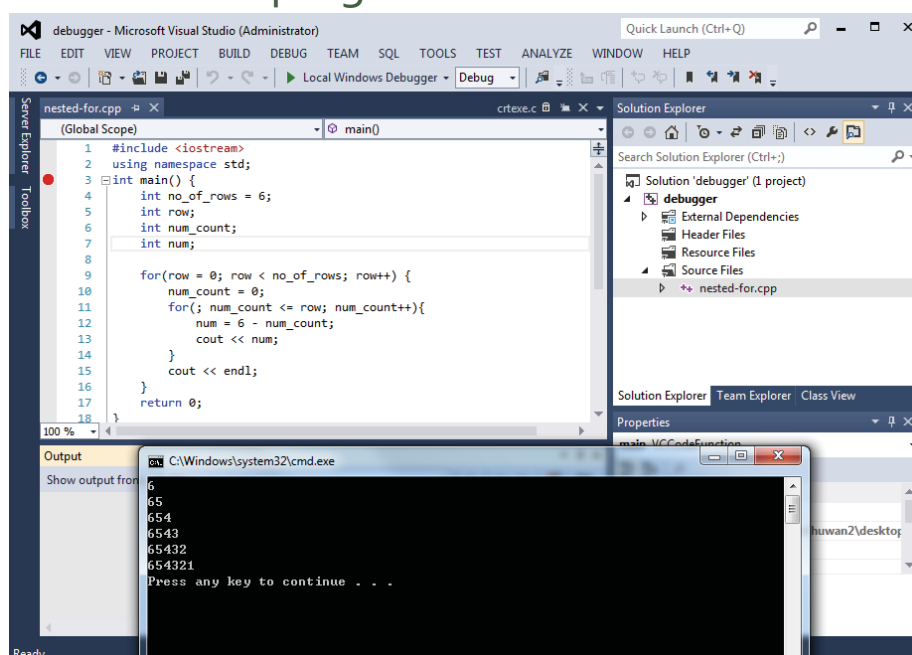
Exercise 2: Wrong Output



29

Exercise 3

- Download `nested-for.cpp` and run inside a debugger to trace the execution of the program



30

Exercise 4

- Do this question after you have learnt the topic on arrays
- Download **sumArray.cpp**, correct the logical errors in the program with the use of the debugger

