C/C++ Program Design

LAB 2

CONTENTS

- Objectives
- Knowledge points
- Exercises

1 Objectives

- Learn How to Download and Install Visual Studio
- Master Fundamental Data types
- Master Arithmetic Operators and Assignment Operators
- Master Keyboard Input and Terminal Output

2 Knowledge Points

2.1 Download and Install Visual Studio

2.2 Fundamental Data Types

2.3 Arithmetic Operators and Assignment Operators

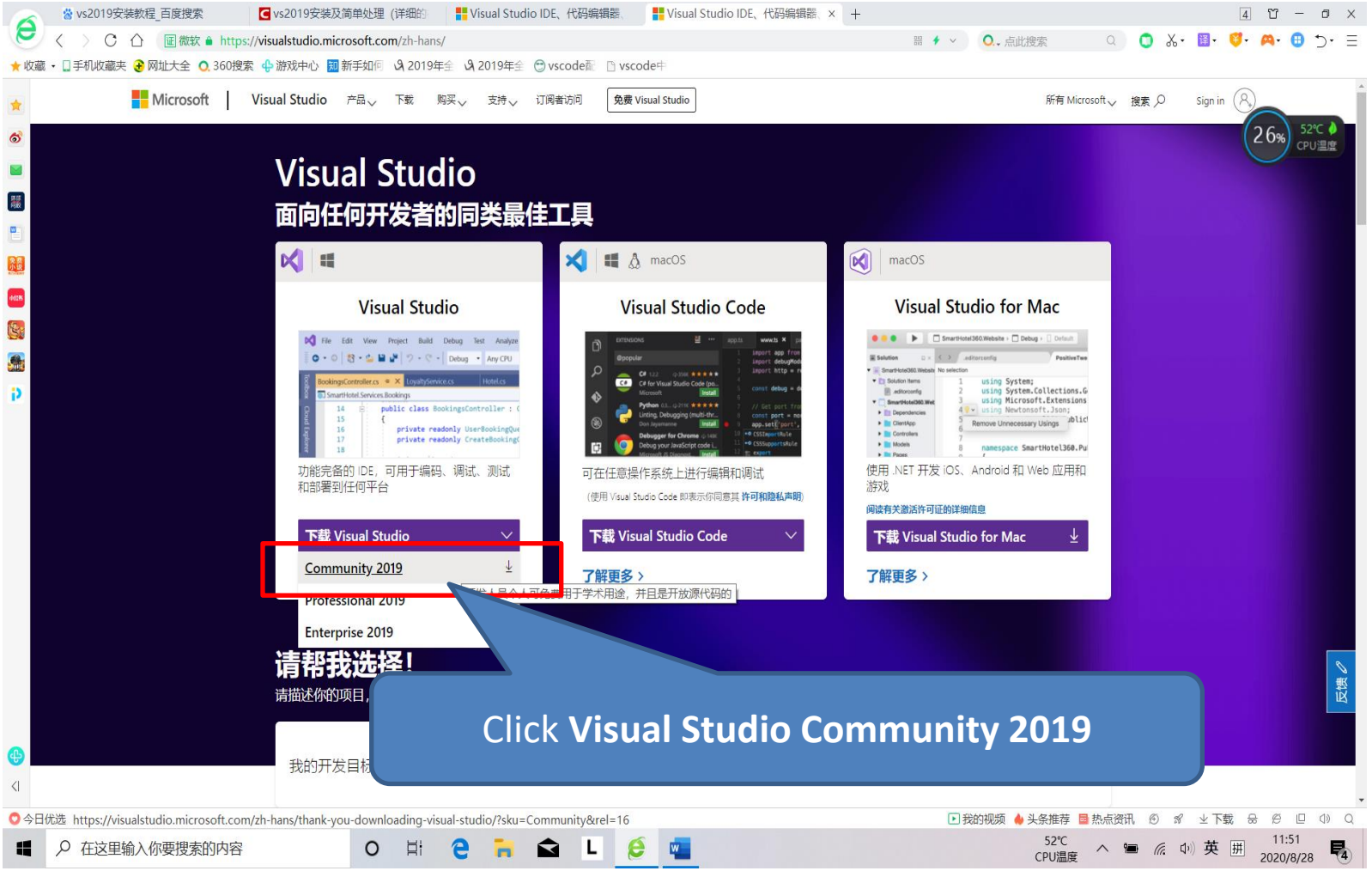
2.4 Input and Output

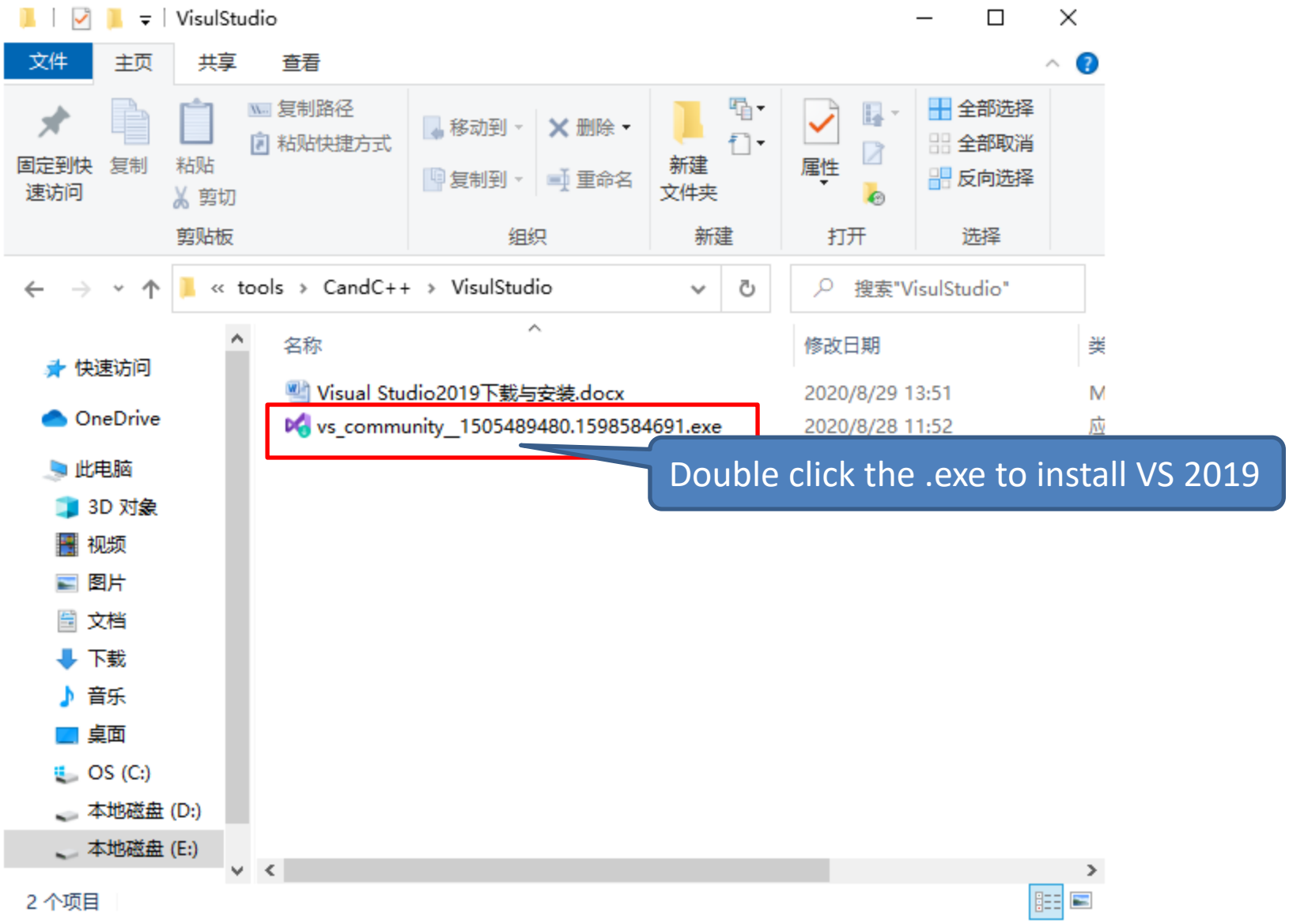
2.1 Download and Install Visual Studio

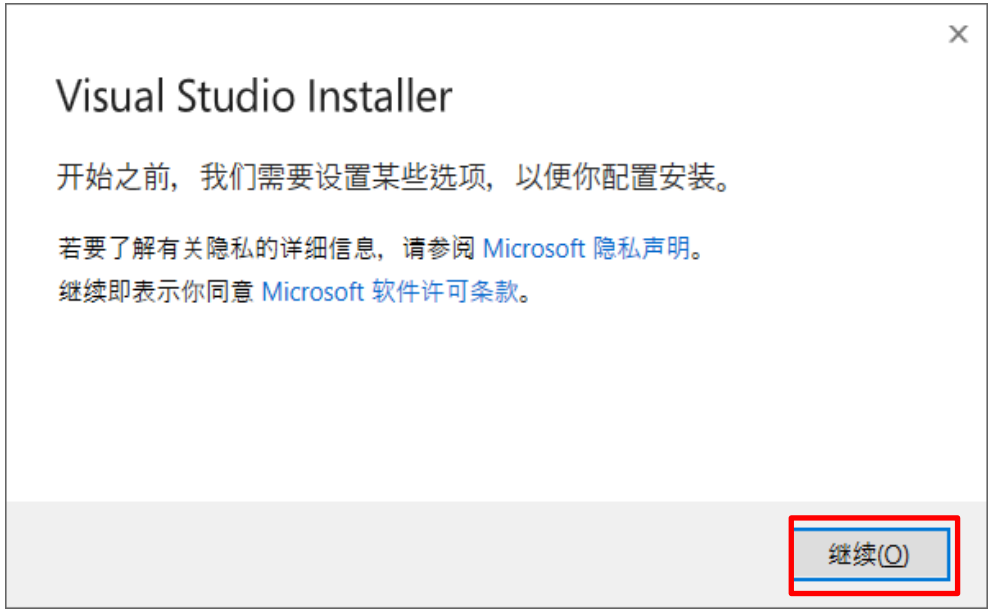
- We have told you how to install VSCode in Lab1.
- As VSCode has already provided predominant function for our programming. So it is enough for you to use.
- However, if you want to use more advanced features on Windows (10),you can install Visual Studio, too.

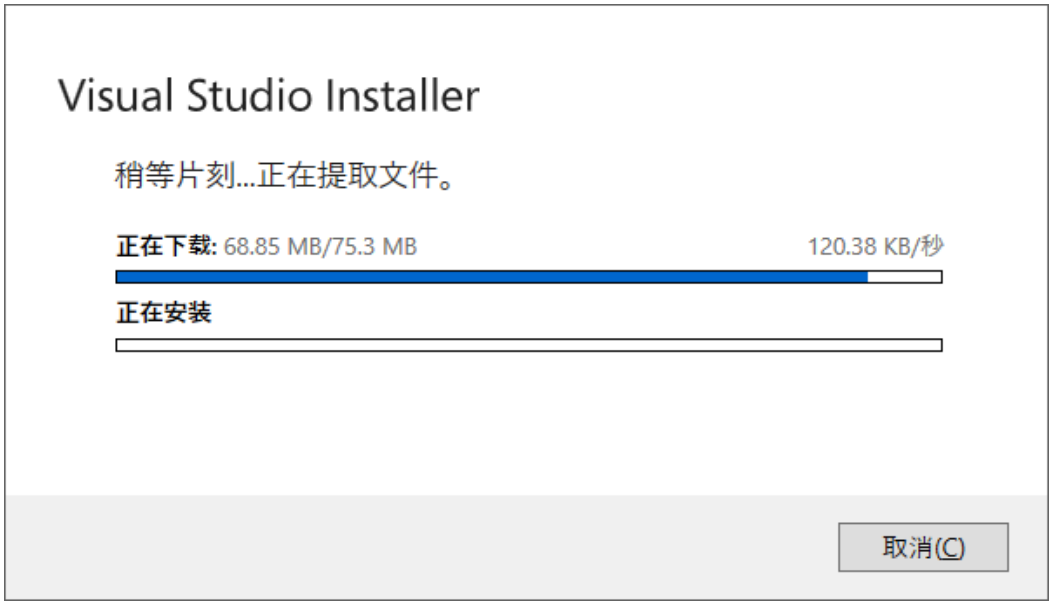
Download and install Visual Studio 2019

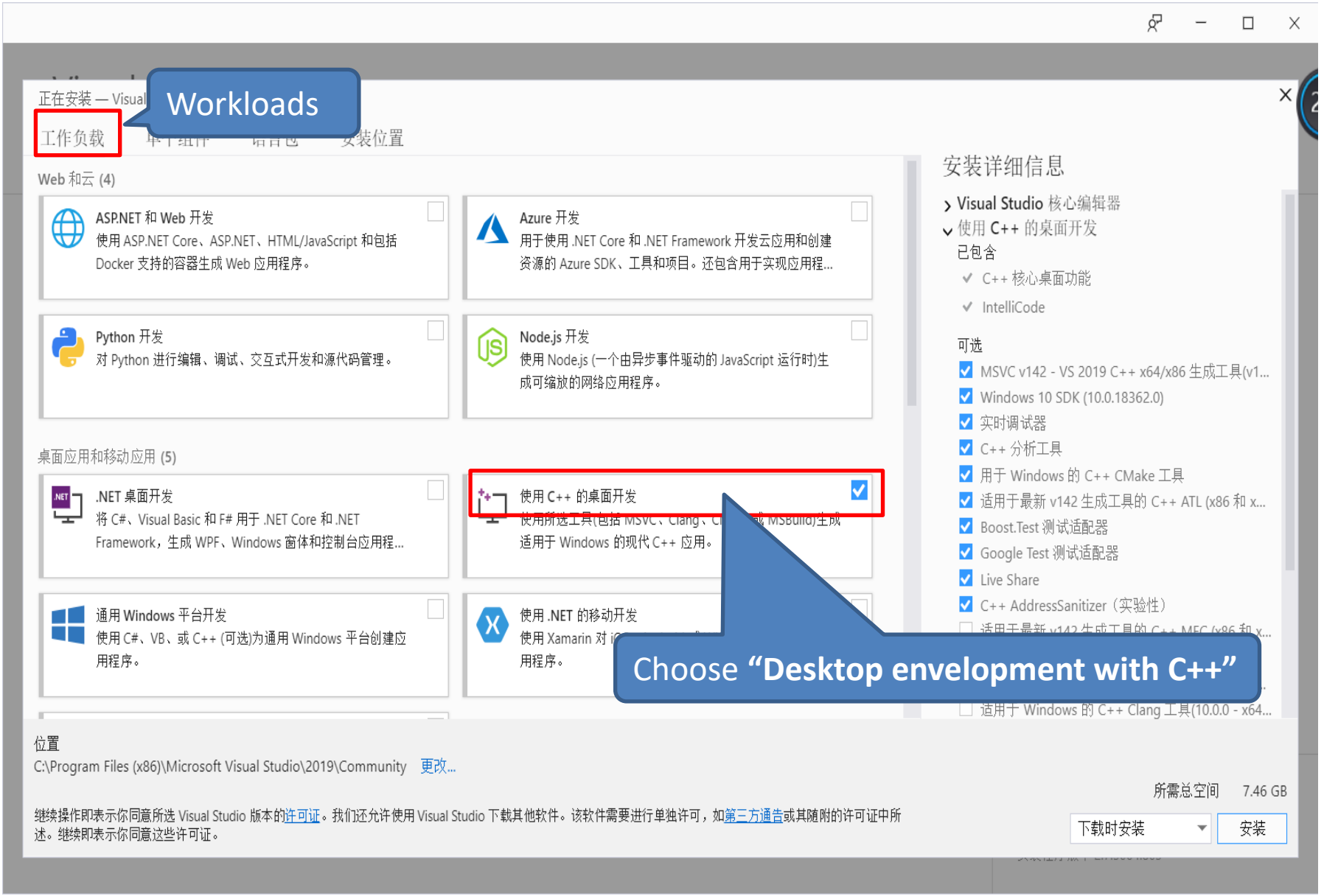
Download url: https://visualstudio.microsoft.com/zh-hans/

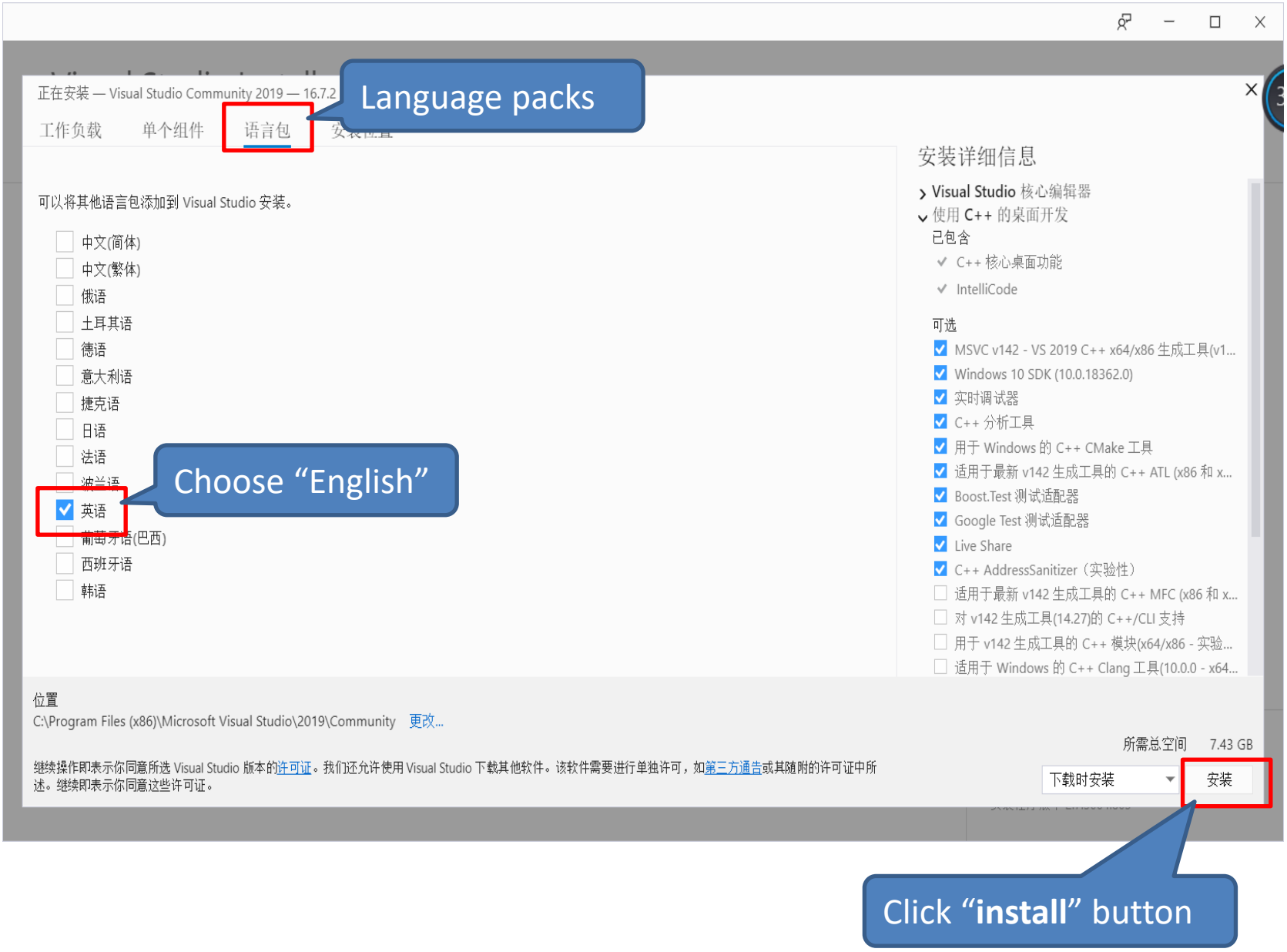



VisulStudio
文件
主页
共享
查看
固定到快速访问
复制
粘贴
剪切
复制路径
粘贴快捷方式
剪贴板
移动到
删除
复制到
重命名
组织
新建文件夹
新建
属性
打开
全部选择
全部取消
反向选择
选择
« tools › CandC++ › VisulStudio
搜索"VisulStudio"
名称
修改日期
快速访问
OneDrive
此电脑
3D 对象
视频
图片
文档
下载
音乐
桌面
OS (C:)
本地磁盘 (D:)
本地磁盘 (E:)
Visual Studio2019下载与安装.docx
2020/8/29 13:51
vs_community__1505489480.1598584691.exe
2020/8/28 11:52
Double click the .exe to install VS 2019
2 个项目

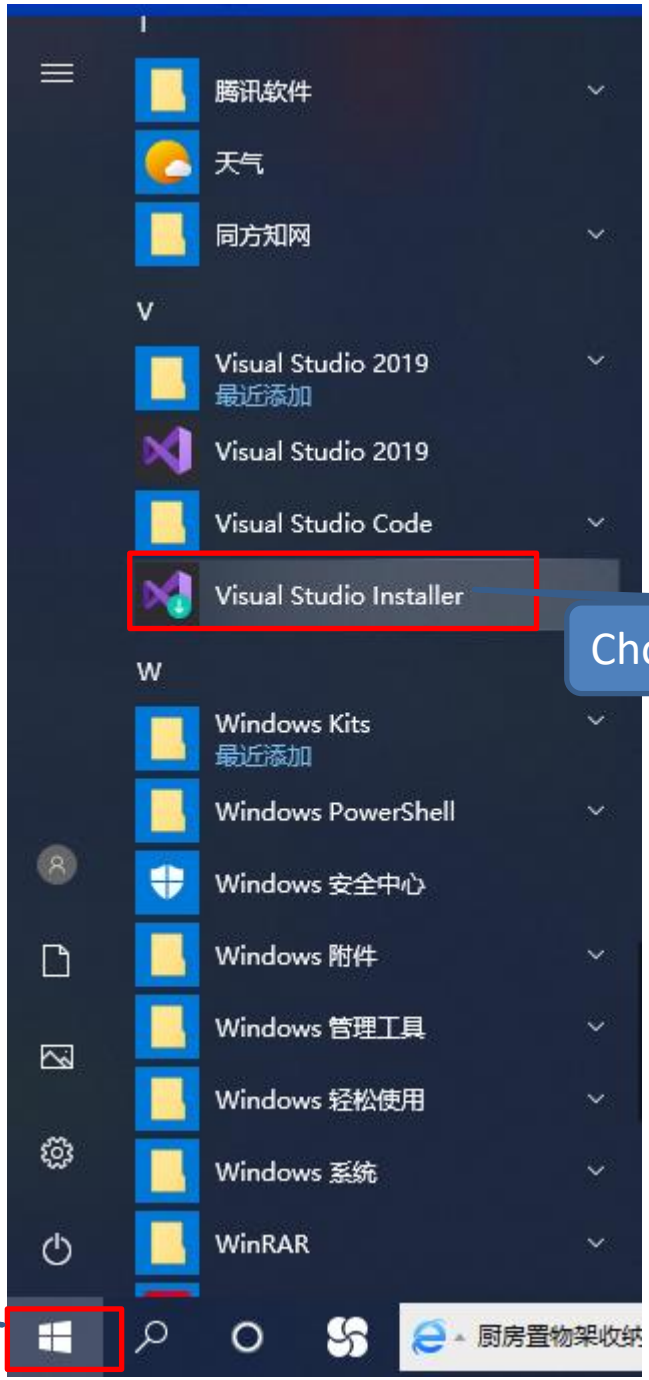

Visual Studio Installer
开始之前，我们需要设置某些选项，以便你配置安装。
若要了解有关隐私的详细信息，请参阅 Microsoft 隐私声明。
继续即表示你同意 Microsoft 软件许可条款。
继续(O)


Visual Studio Installer
稍等片刻...正在提取文件。
正在下载: 68.85 MB/75.3 MB
120.38 KB/秒
正在安装
取消(C)


Workloads
正在安装 — Visual
工作负载
单个组件
语言包
安装位置
Web 和云 (4)
ASP.NET 和 Web 开发
使用 ASP.NET Core、ASP.NET、HTML/JavaScript 和包括 Docker 支持的容器生成 Web 应用程序。
Azure 开发
用于使用 .NET Core 和 .NET Framework 开发云应用和创建资源的 Azure SDK、工具和项目。还包含用于实现应用程...
Python 开发
对 Python 进行编辑、调试、交互式开发和源代码管理。
Node.js 开发
使用 Node.js (一个由异步事件驱动的 JavaScript 运行时)生成可缩放的网络应用程序。
桌面应用和移动应用 (5)
.NET 桌面开发
将 C#、Visual Basic 和 F# 用于 .NET Core 和 .NET Framework，生成 WPF、Windows 窗体和控制台应用程...
使用 C++ 的桌面开发
适用于 Windows 的现代 C++ 应用。
通用 Windows 平台开发
使用 C#、VB、或 C++ (可选)为通用 Windows 平台创建应用程序。
使用 .NET 的移动开发
使用 Xamarin 对
用程序。
安装详细信息
Visual Studio 核心编辑器
使用 C++ 的桌面开发
已包含
C++ 核心桌面功能
IntelliCode
可选
MSVC v142 - VS 2019 C++ x64/x86 生成工具(v1...
Windows 10 SDK (10.0.18362.0)
实时调试器
C++ 分析工具
用于 Windows 的 C++ CMake 工具
适用于最新 v142 生成工具的 C++ ATL (x86 和 x...
Boost.Test 测试适配器
Google Test 测试适配器
Live Share
C++ AddressSanitizer（实验性）
适用于 Windows 的 C++ Clang 工具(10.0.0 - x64...
Choose “Desktop envelopment with C++”
位置
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community
更改...
所需总空间
7.46 GB
继续操作即表示你同意所选 Visual Studio 版本的许可证。我们还允许使用 Visual Studio 下载其他软件。该软件需要进行单独许可，如第三方通告或其随附的许可证中所述。继续即表示你同意这些许可证。
下载时安装
安装

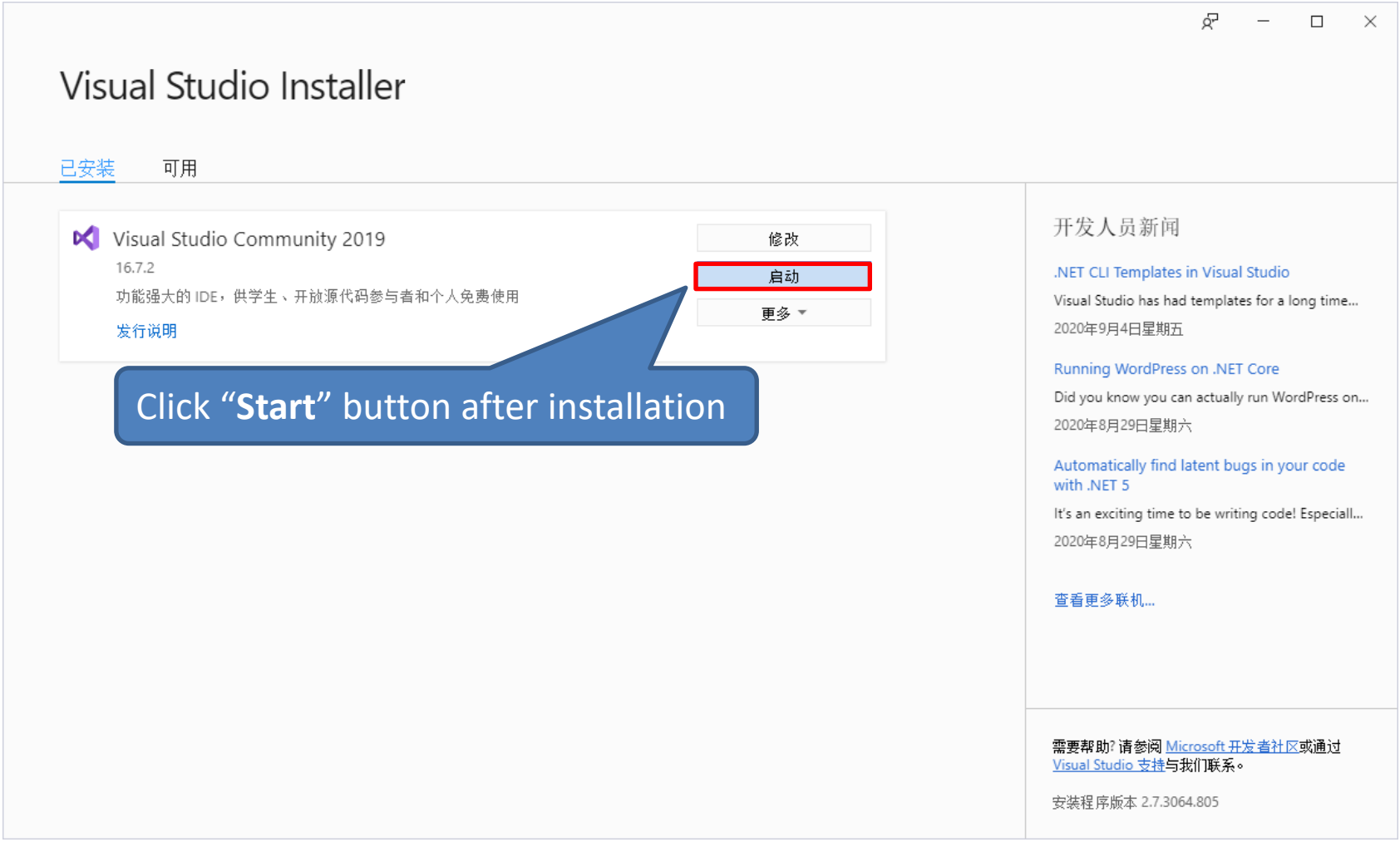

Language packs
正在安装 — Visual Studio Community 2019 — 16.7.2
工作负载
单个组件
语言包
安装位置
可以将其他语言包添加到 Visual Studio 安装。
中文(简体)
中文(繁体)
俄语
土耳其语
德语
意大利语
捷克语
日语
法语
波兰语
英语
葡萄牙语(巴西)
西班牙语
韩语
Choose “English”
安装详细信息
Visual Studio 核心编辑器
使用 C++ 的桌面开发
已包含
C++ 核心桌面功能
IntelliCode
可选
MSVC v142 - VS 2019 C++ x64/x86 生成工具(v1...
Windows 10 SDK (10.0.18362.0)
实时调试器
C++ 分析工具
用于 Windows 的 C++ CMake 工具
适用于最新 v142 生成工具的 C++ ATL (x86 和 x...
Boost.Test 测试适配器
Google Test 测试适配器
Live Share
C++ AddressSanitizer（实验性）
适用于最新 v142 生成工具的 C++ MFC (x86 和 x...
对 v142 生成工具(14.27)的 C++/CLI 支持
用于 v142 生成工具的 C++ 模块(x64/x86 - 实验...
适用于 Windows 的 C++ Clang 工具(10.0.0 - x64...
位置
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community
更改...
所需总空间 7.43 GB
继续操作即表示你同意所选 Visual Studio 版本的许可证。我们还允许使用 Visual Studio 下载其他软件。该软件需要进行单独许可，如第三方通告或其随附的许可证中所述。继续即表示你同意这些许可证。
下载时安装
安装
Click “install” button

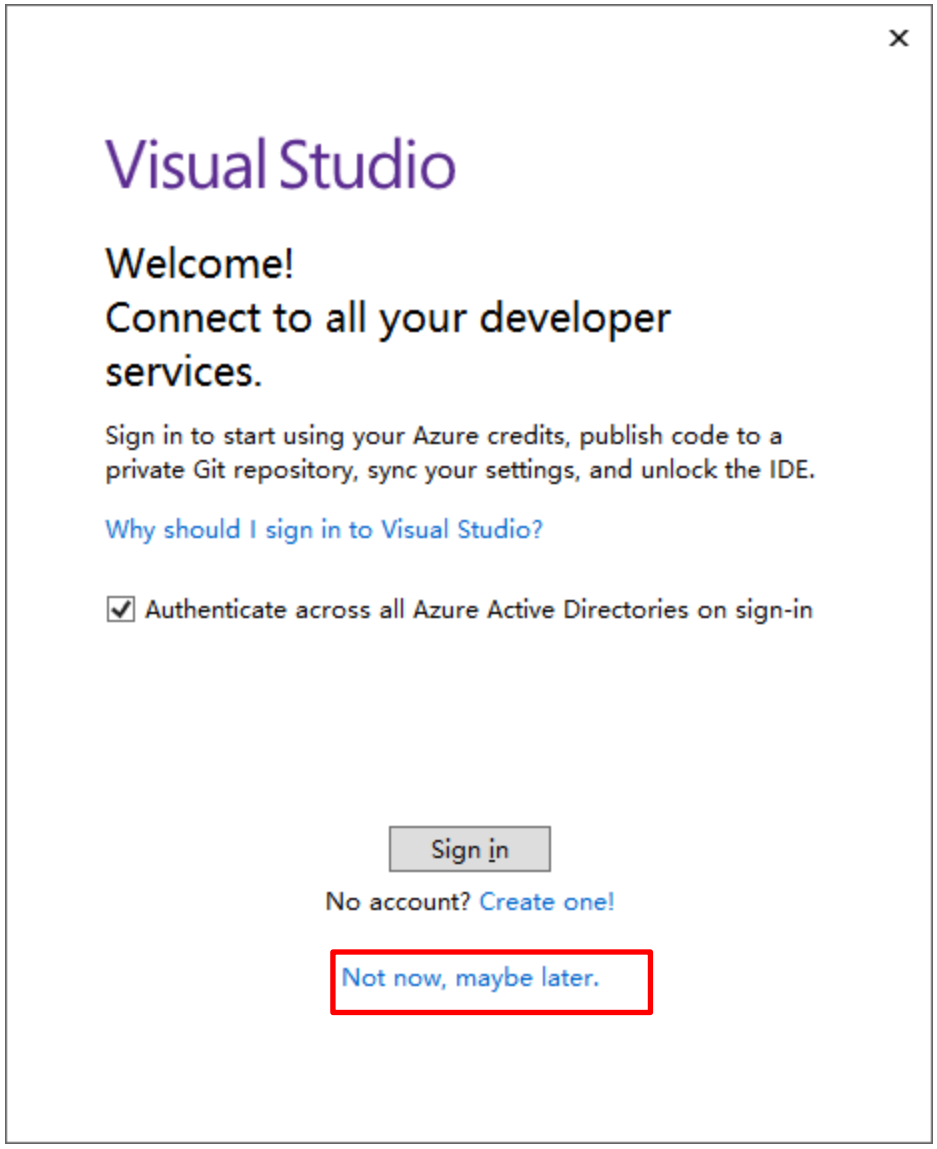

Visual Studio Installer
已安装
可用
Visual Studio Community 2019
16.7.2
重启
需要重启。若需要，则重启后将自动恢复所有剩余的设置。
需要重启
成功了! 但还有一步，请在启动 Visual Studio Community 2019 前重启计算机。
获取疑难解答使用技巧
重启
以后再说
Restart your computer
开发人员新闻
.NET CLI Templates in Visual Studio
Visual Studio has had templates for a long time...
2020年9月4日星期五
Running WordPress on .NET Core
Did you know you can actually run WordPress on...
2020年8月29日星期六
Automatically find latent bugs in your code with .NET 5
It's an exciting time to be writing code! Especiall...
2020年8月29日星期六
查看更多联机...
需要帮助? 请参阅 Microsoft 开发者社区或通过 Visual Studio 支持与我们联系。
安装程序版本 2.7.3064.805

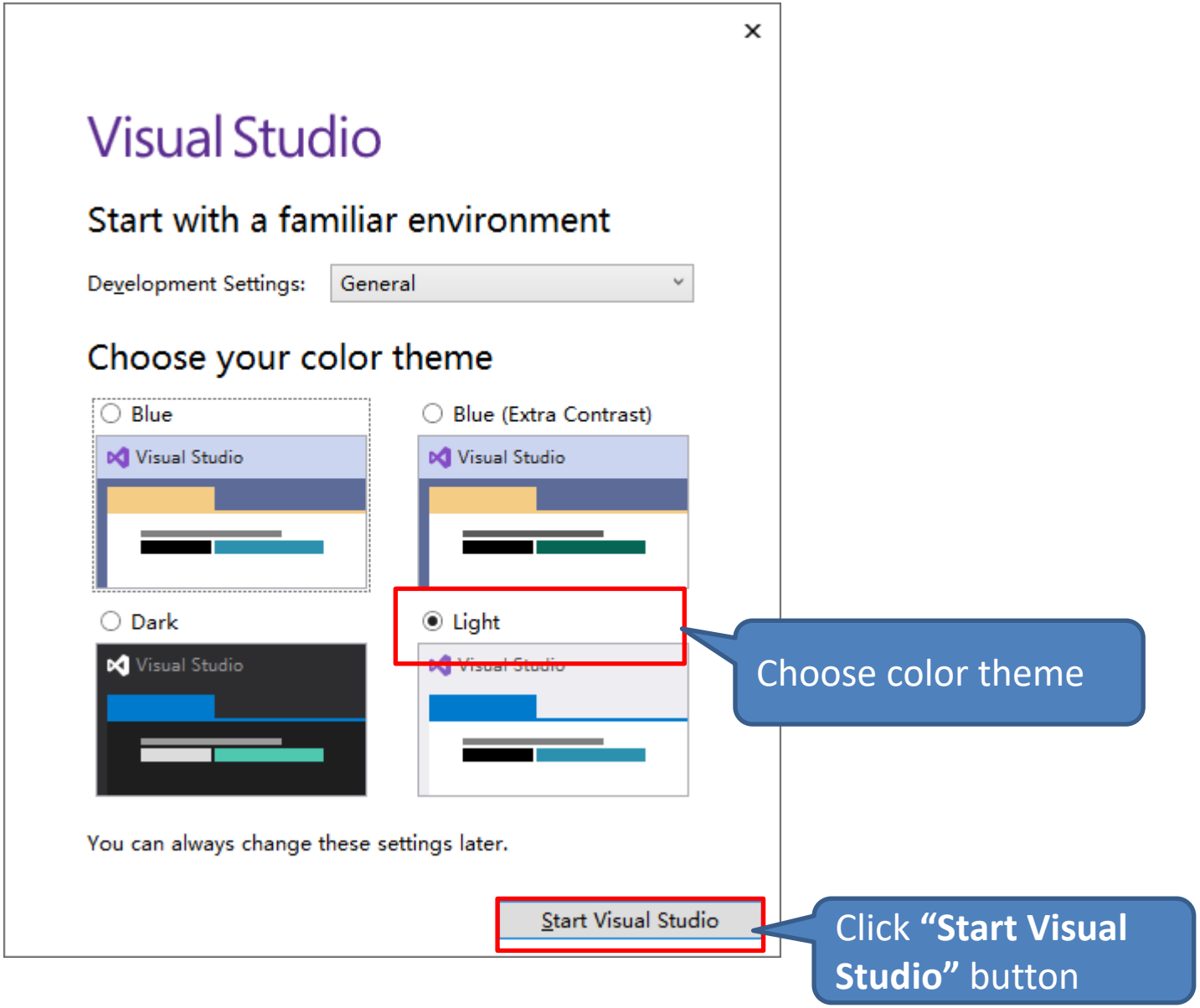


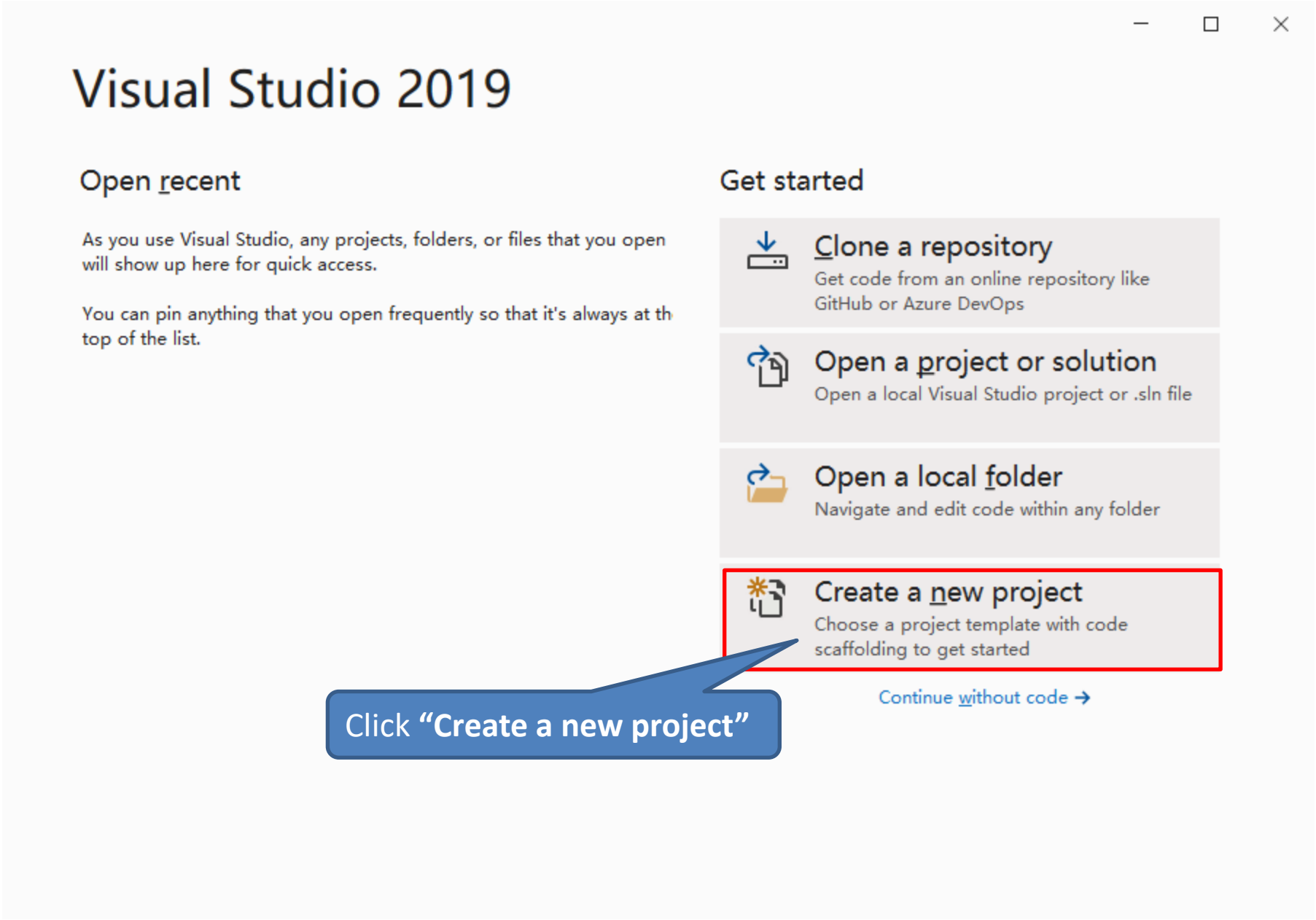
Choose “**Visual Studio Installer**”

Click the **Start menu**

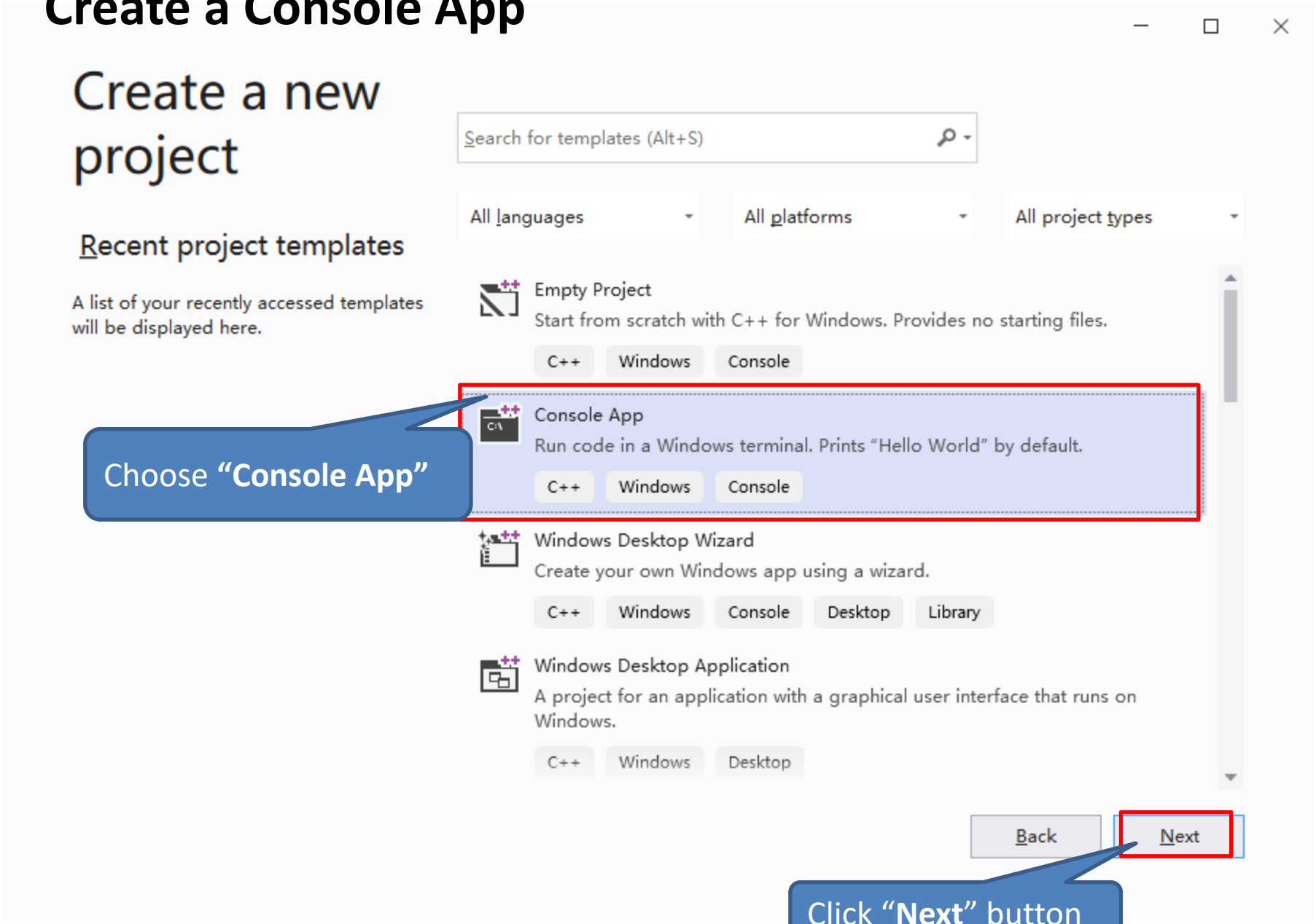

Visual Studio Installer
已安装
可用
Visual Studio Community 2019
16.7.2
功能强大的 IDE，供学生、开放源代码参与者和个人免费使用
发行说明
修改
启动
更多
Click “Start” button after installation
开发人员新闻
.NET CLI Templates in Visual Studio
Visual Studio has had templates for a long time...
2020年9月4日星期五
Running WordPress on .NET Core
Did you know you can actually run WordPress on...
2020年8月29日星期六
Automatically find latent bugs in your code with .NET 5
It's an exciting time to be writing code! Especiall...
2020年8月29日星期六
查看更多联机...
需要帮助? 请参阅 Microsoft 开发者社区或通过 Visual Studio 支持与我们联系。
安装程序版本 2.7.3064.805

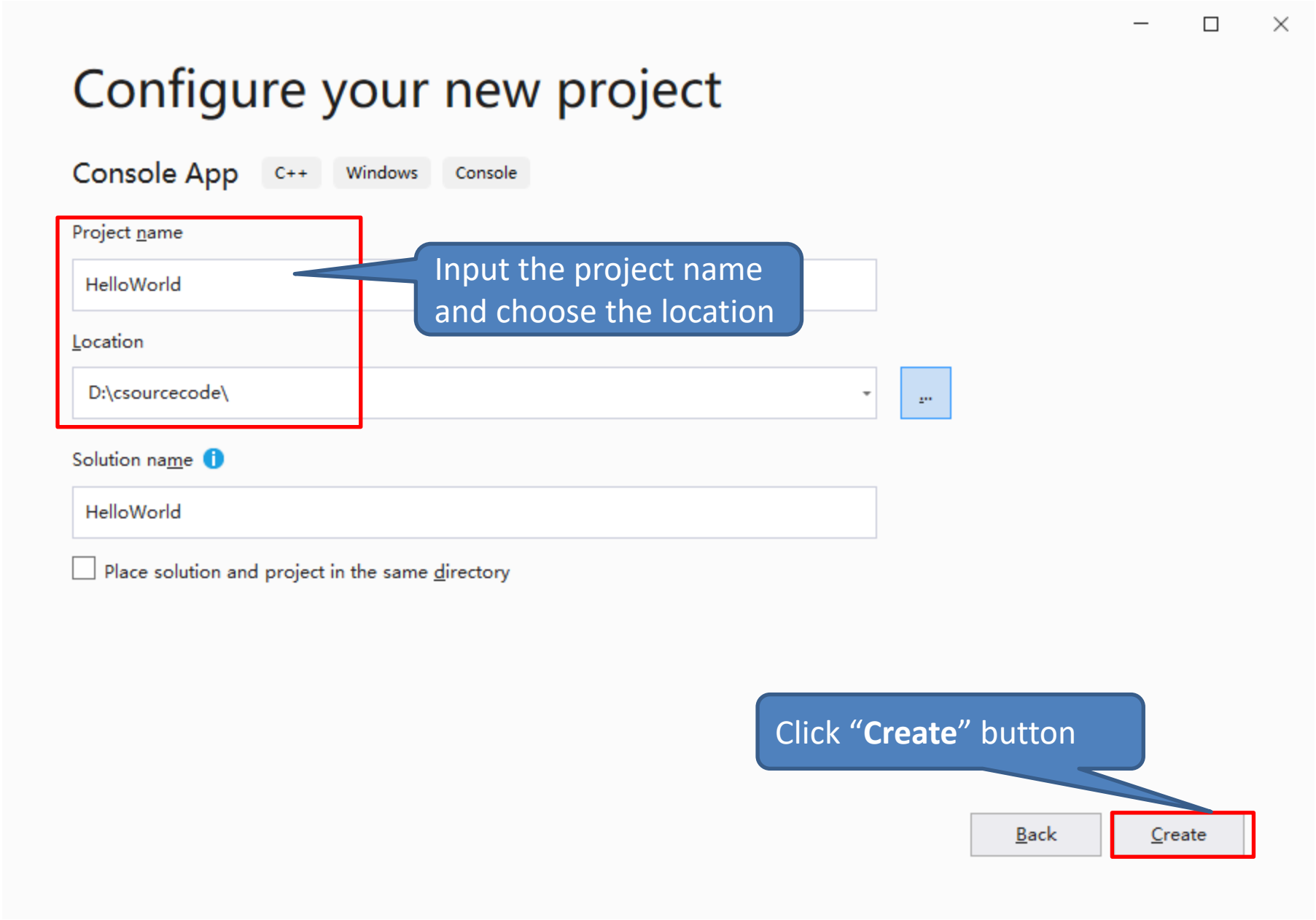

Visual Studio
Welcome!
Connect to all your developer services.
Sign in to start using your Azure credits, publish code to a private Git repository, sync your settings, and unlock the IDE.
Why should I sign in to Visual Studio?
Authenticate across all Azure Active Directories on sign-in
Sign in
No account? Create one!
Not now, maybe later.

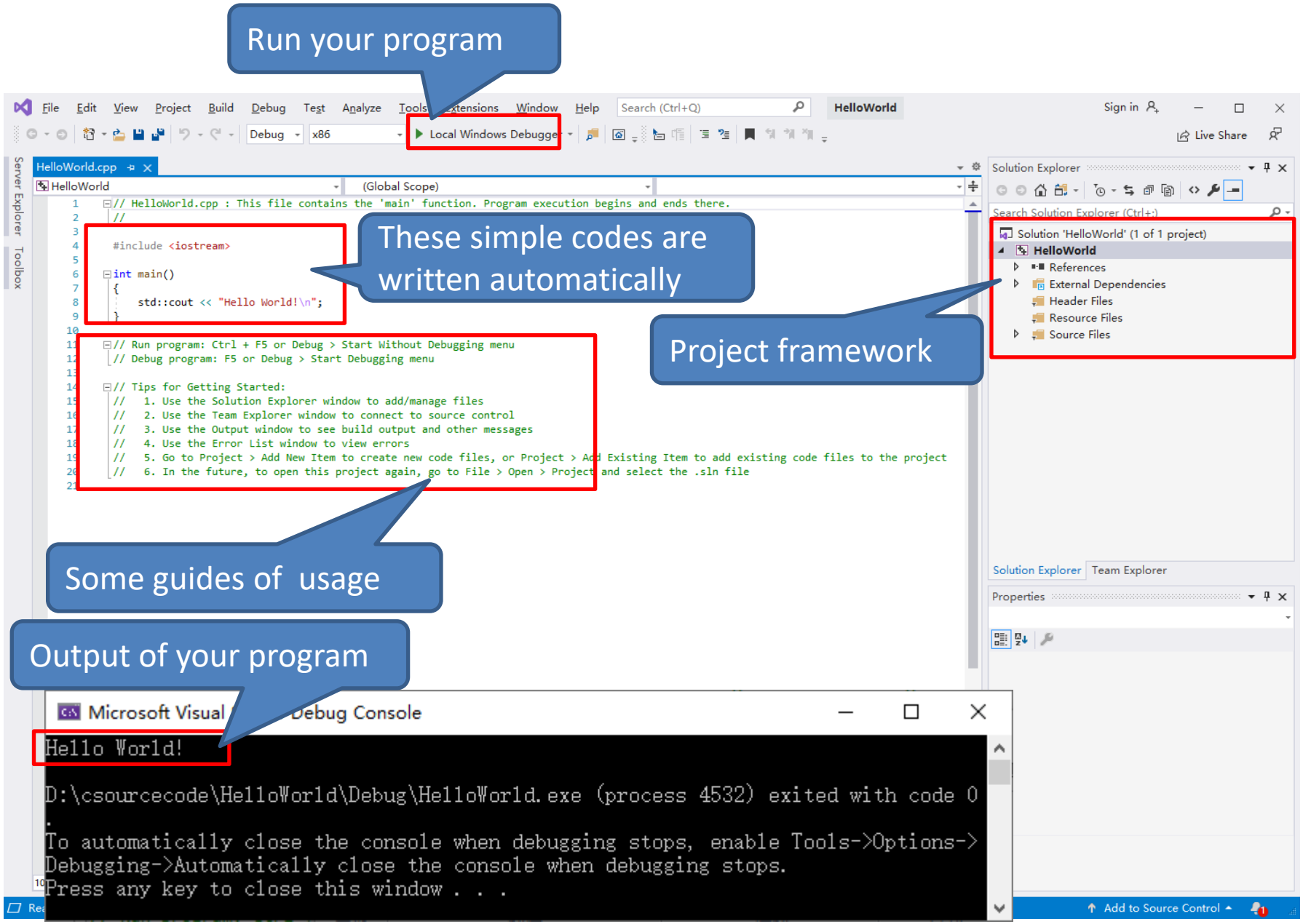

Visual Studio
Start with a familiar environment
Development Settings:
General
Choose your color theme
Blue
Visual Studio
Blue (Extra Contrast)
Visual Studio
Dark
Visual Studio
Light
Visual Studio
You can always change these settings later.
Start Visual Studio
Choose color theme
Click "Start Visual Studio" button


Visual Studio 2019
Open recent
As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.
You can pin anything that you open frequently so that it's always at th top of the list.
Get started
Clone a repository
Get code from an online repository like GitHub or Azure DevOps
Open a project or solution
Open a local Visual Studio project or .sln file
Open a local folder
Navigate and edit code within any folder
Create a new project
Choose a project template with code scaffolding to get started
Continue without code →
Click "Create a new project"

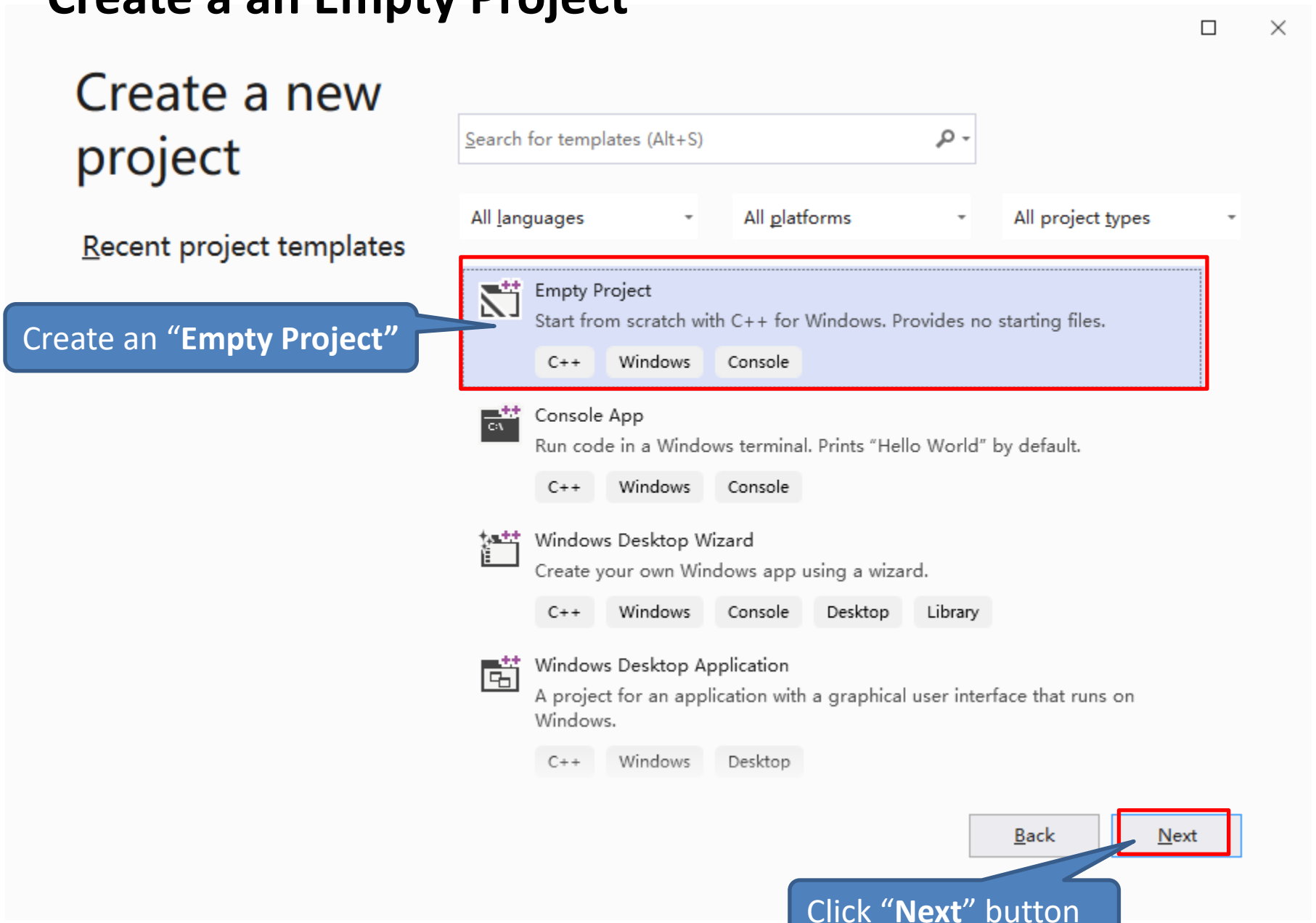
Create a Console App

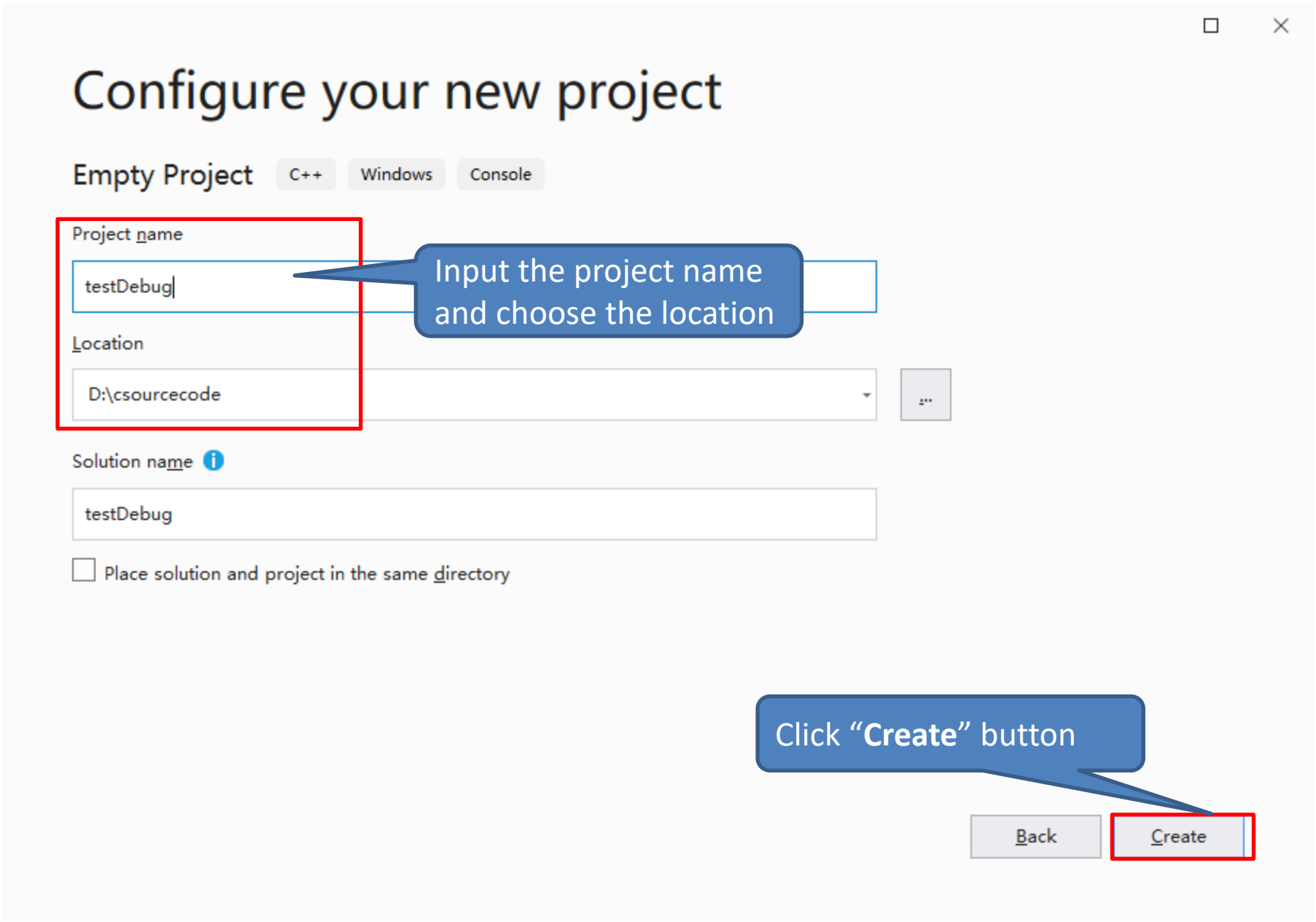


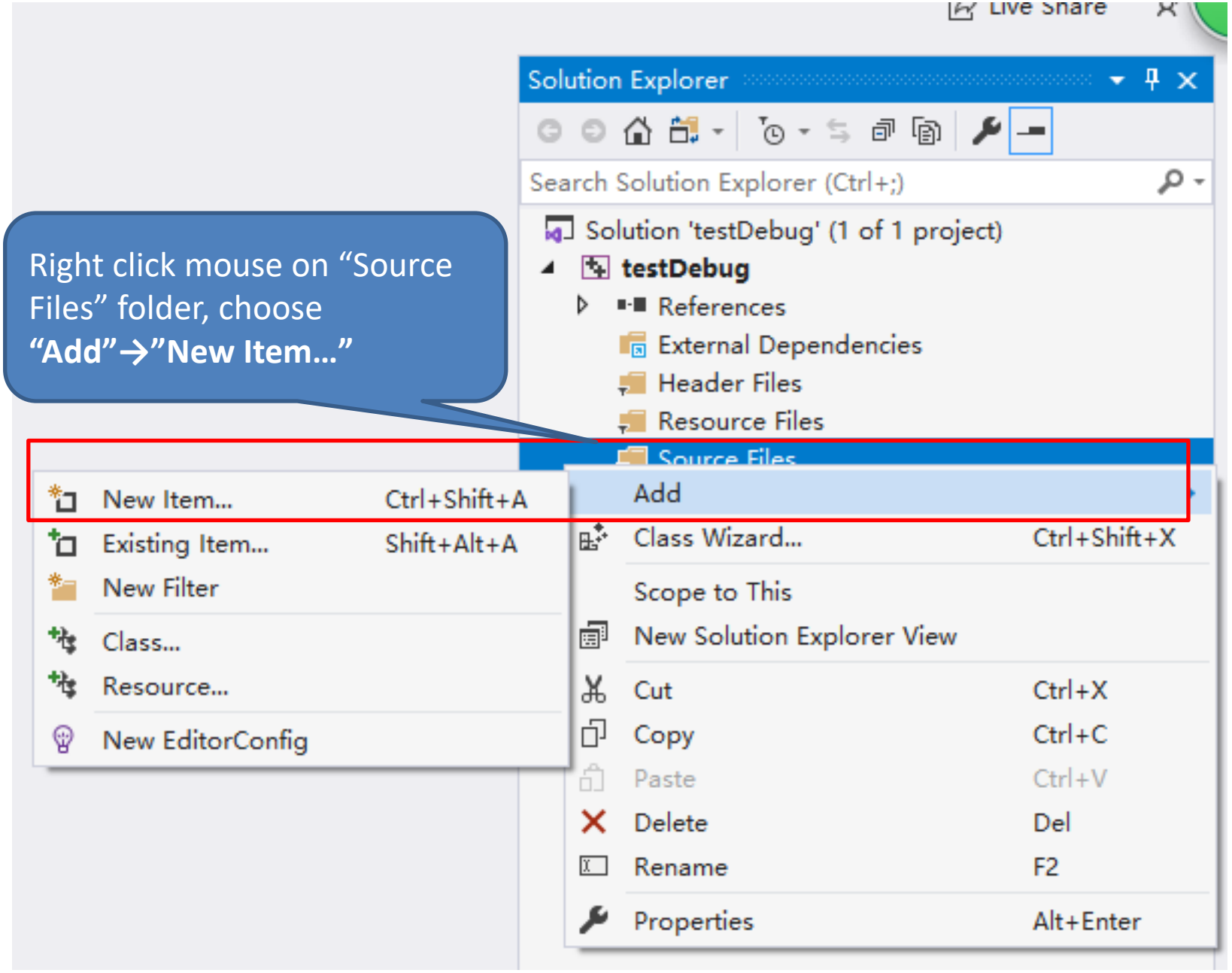

Configure your new project
Console App
C++
Windows
Console
Project name
HelloWorld
Input the project name and choose the location
Location
D:\csourcecode\
...
Solution name
HelloWorld
Place solution and project in the same directory
Click "Create" button
Back
Create

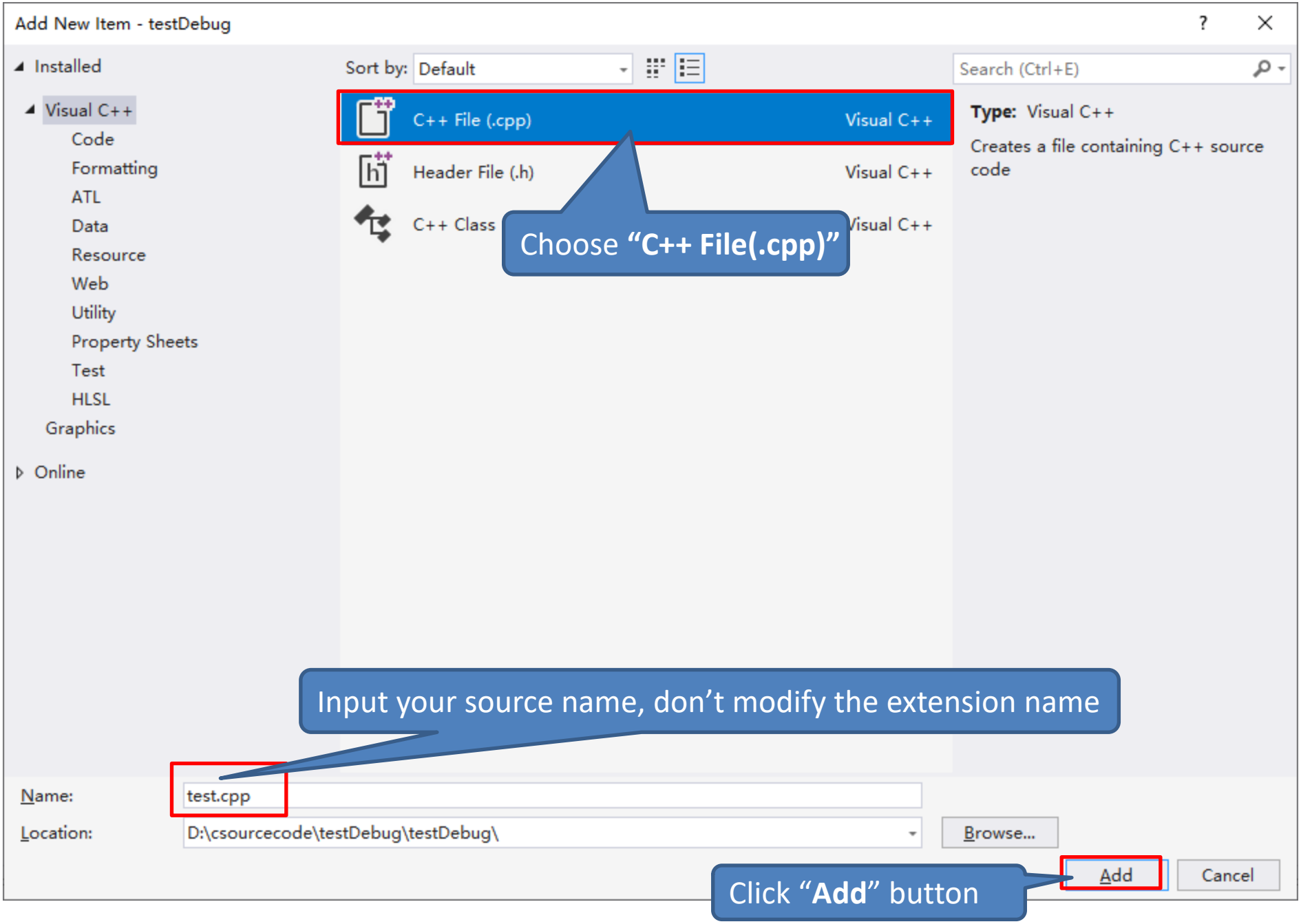

Run your program
File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
HelloWorld
Sign in
Debug
x86
Local Windows Debugger
Live Share
HelloWorld.cpp
HelloWorld
(Global Scope)
// HelloWorld.cpp : This file contains the 'main' function. Program execution begins and ends there.
//
#include <iostream>
int main()
{
std::cout << "Hello World!\n";
}
// Run program: Ctrl + F5 or Debug > Start Without Debugging menu
// Debug program: F5 or Debug > Start Debugging menu
// Tips for Getting Started:
// 1. Use the Solution Explorer window to add/manage files
// 2. Use the Team Explorer window to connect to source control
// 3. Use the Output window to see build output and other messages
// 4. Use the Error List window to view errors
// 5. Go to Project > Add New Item to create new code files, or Project > Add Existing Item to add existing code files to the project
// 6. In the future, to open this project again, go to File > Open > Project and select the .sln file
These simple codes are written automatically
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'HelloWorld' (1 of 1 project)
HelloWorld
References
External Dependencies
Header Files
Resource Files
Source Files
Project framework
Solution Explorer
Team Explorer
Properties
Some guides of usage
Output of your program
Microsoft Visual Debug Console
Hello World!
D:\csourcecode\HelloWorld\Debug\HelloWorld.exe (process 4532) exited with code 0
.
To automatically close the console when debugging stops, enable Tools->Options->
Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
Add to Source Control

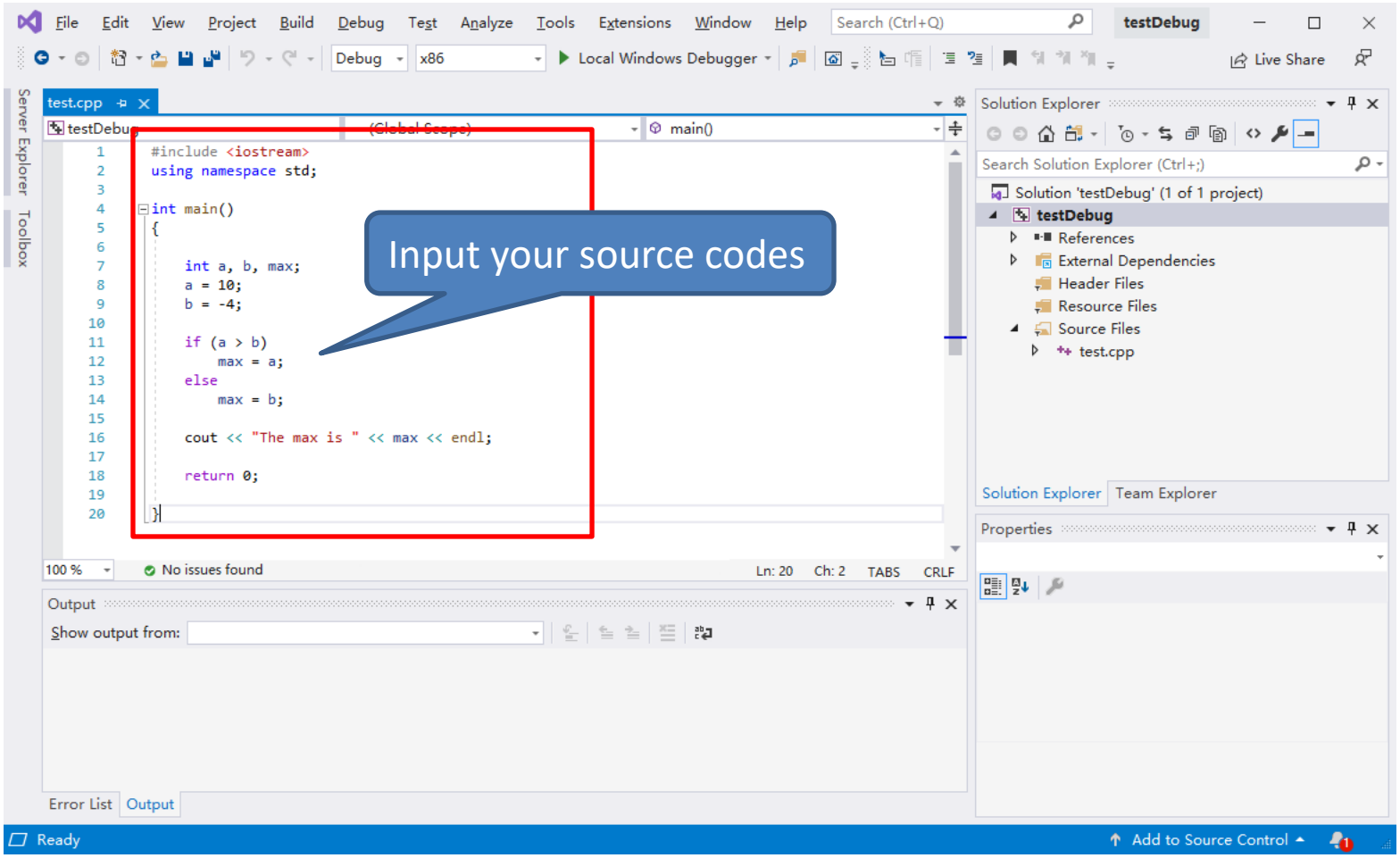
Create a an Empty Project



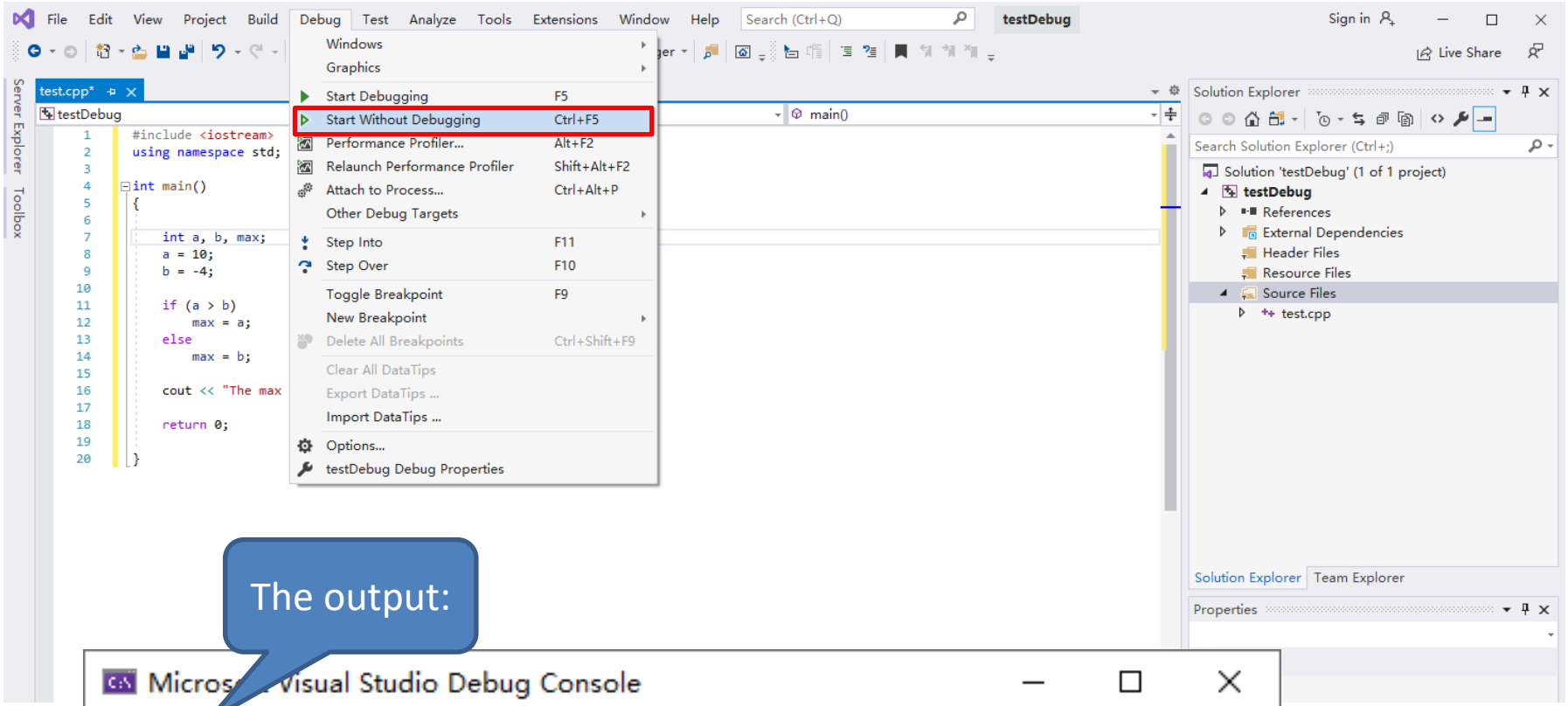

Configure your new project
Empty Project
C++
Windows
Console
Project name
testDebug
Input the project name and choose the location
Location
D:\csourcecode
...
Solution name
testDebug
Place solution and project in the same directory
Click “Create” button
Back
Create


Live Share
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'testDebug' (1 of 1 project)
testDebug
References
External Dependencies
Header Files
Resource Files
Source Files
Right click mouse on “Source Files” folder, choose “Add”→”New Item…”
New Item... Ctrl+Shift+A
Existing Item... Shift+Alt+A
New Filter
Class...
Resource...
New EditorConfig
Add
Class Wizard... Ctrl+Shift+X
Scope to This
New Solution Explorer View
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Rename F2
Properties Alt+Enter

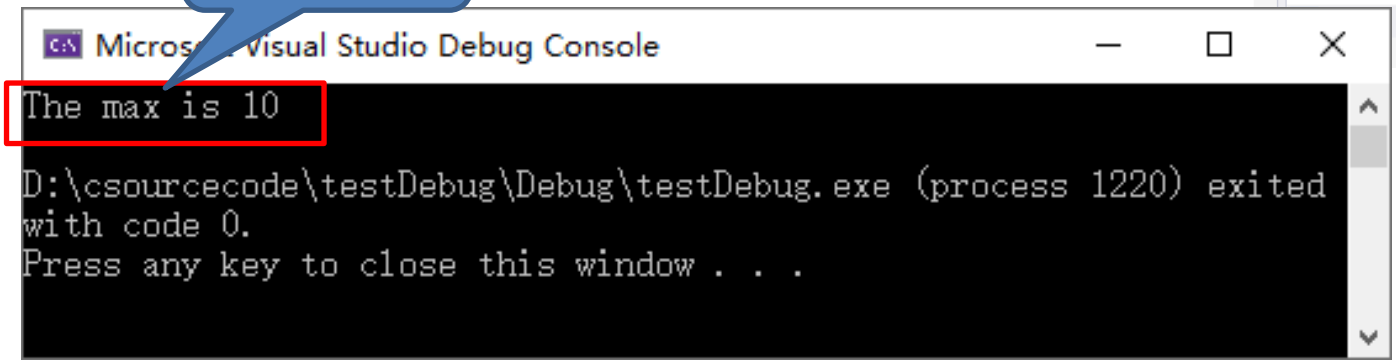

Add New Item - testDebug
Installed
Visual C++
Code
Formatting
ATL
Data
Resource
Web
Utility
Property Sheets
Test
HLSL
Graphics
Online
Sort by: Default
Search (Ctrl+E)
C++ File (.cpp) Visual C++
Header File (.h) Visual C++
C++ Class Visual C++
Type: Visual C++
Creates a file containing C++ source code
Choose "C++ File(.cpp)"
Input your source name, don't modify the extension name
Name: test.cpp
Location: D:\csourcecode\testDebug\testDebug\
Browse...
Add
Cancel
Click "Add" button


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
testDebug
Debug
x86
Local Windows Debugger
Live Share
test.cpp
main()
#include <iostream>
using namespace std;
int main()
{
int a, b, max;
a = 10;
b = -4;
if (a > b)
max = a;
else
max = b;
cout << "The max is " << max << endl;
return 0;
}
Input your source codes
100 %
No issues found
Ln: 20 Ch: 2 TABS CRLF
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'testDebug' (1 of 1 project)
testDebug
References
External Dependencies
Header Files
Resource Files
Source Files
test.cpp
Solution Explorer Team Explorer
Properties
Output
Show output from:
Error List Output
Ready
Add to Source Control

Choose “Debug”→”Start without Debugging” to run your program.

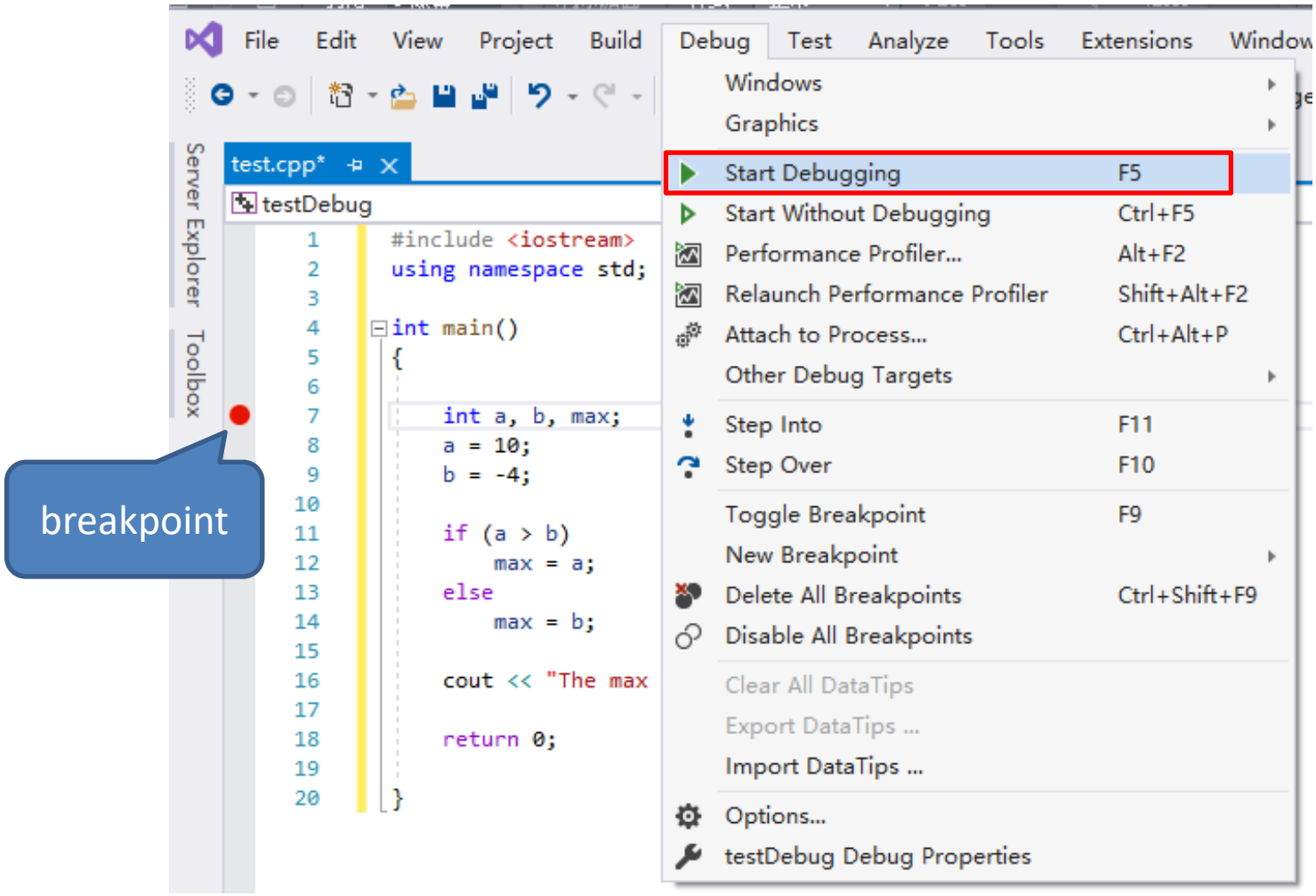


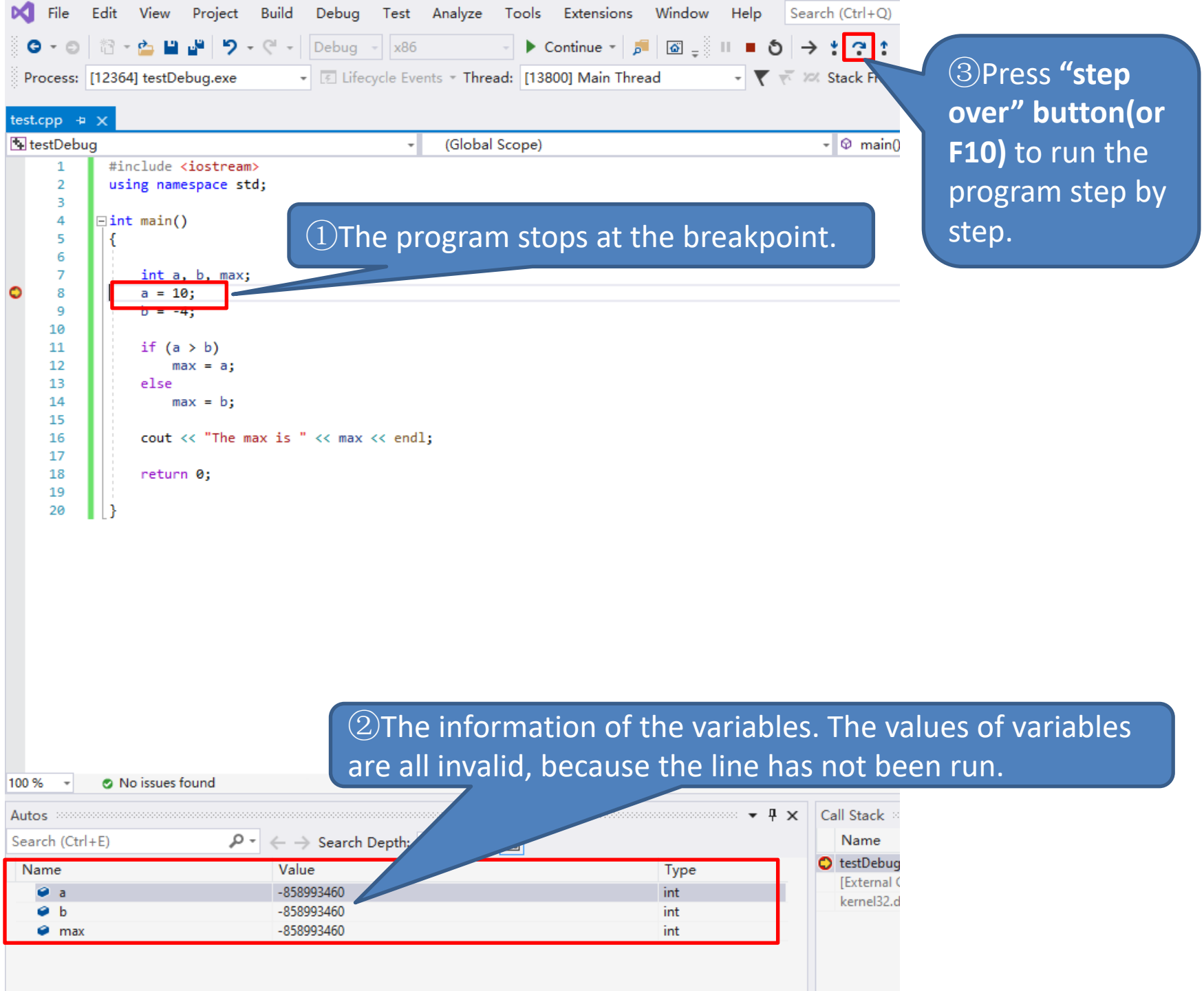
The output:

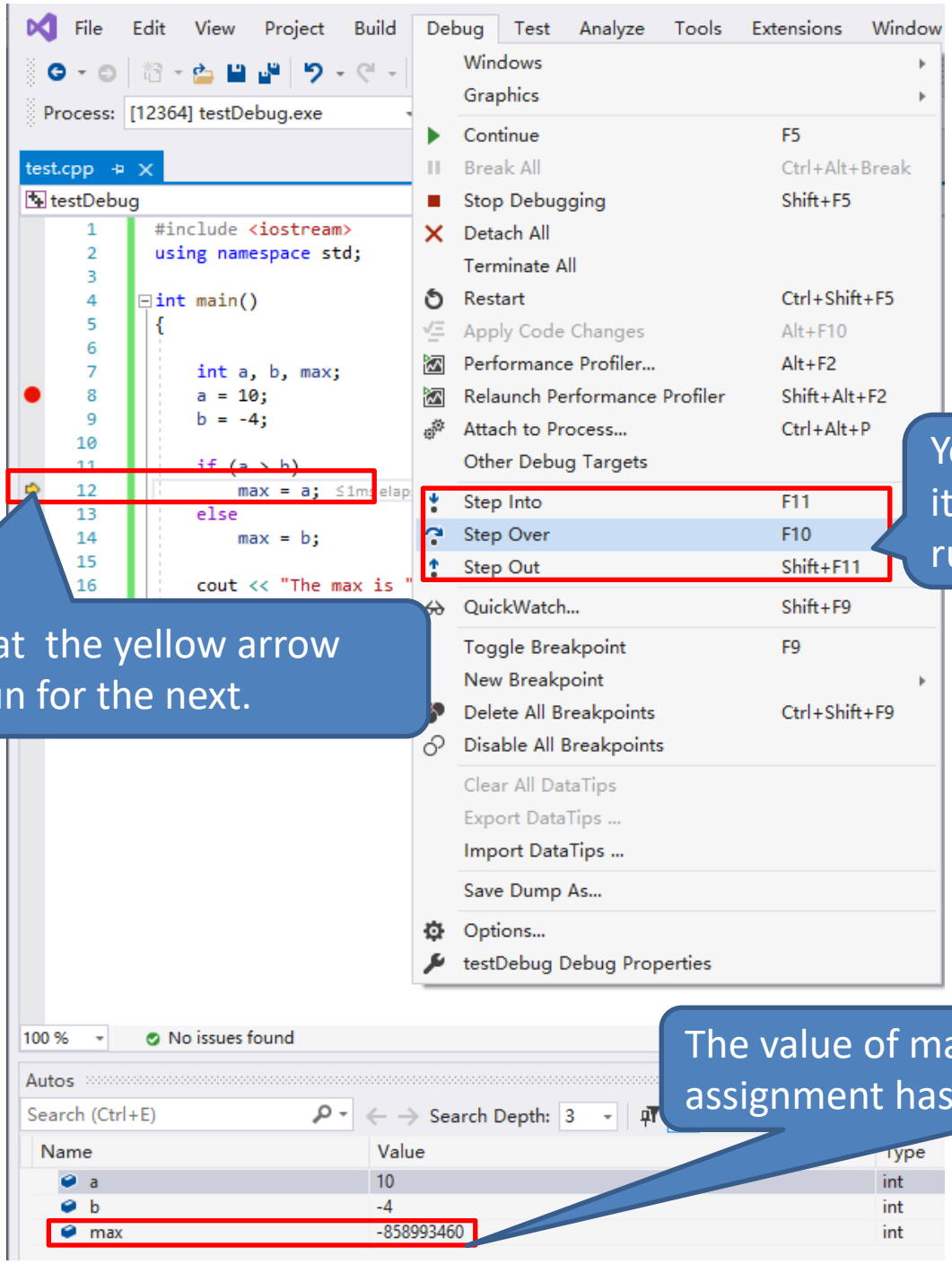


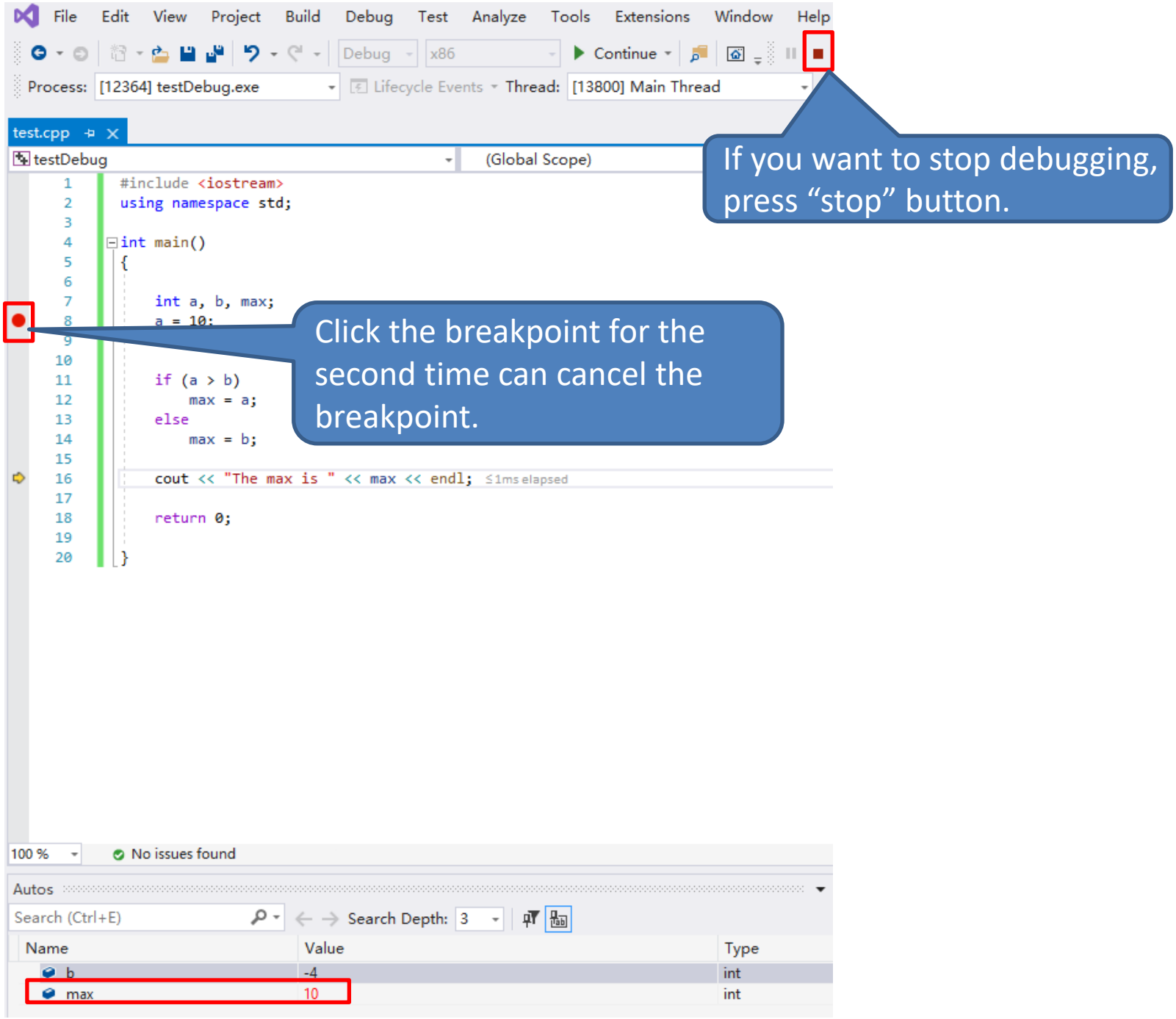
How to debug your program?

Click mouse on the edge of the line on which you want to **set a breakpoint** and choose **“Debug”→”Start Debugging” (or press F5)** to run your program at debugging way.

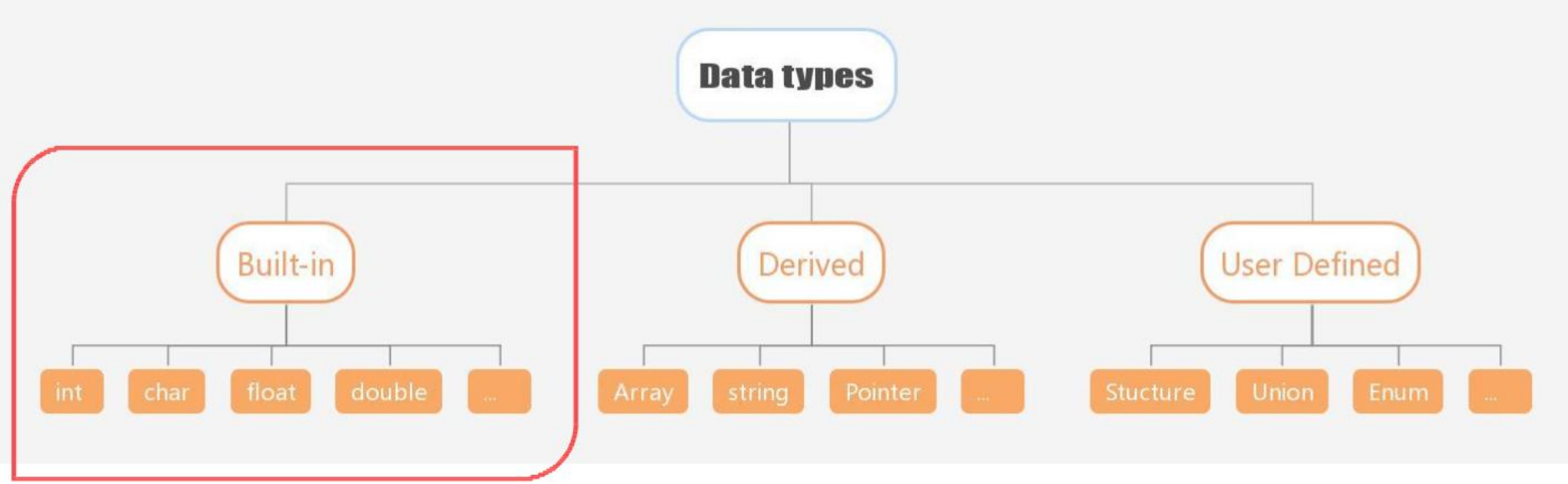



File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Debug
x86
Continue
Process: [12364] testDebug.exe
Lifecycle Events
Thread: [13800] Main Thread
test.cpp
testDebug
(Global Scope)
main()
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6
7 int a, b, max;
8 a = 10;
9 b = -4;
10
11 if (a > b)
12 max = a;
13 else
14 max = b;
15
16 cout << "The max is " << max << endl;
17
18 return 0;
19
20 }
①The program stops at the breakpoint.
③Press “step over” button(or F10) to run the program step by step.
②The information of the variables. The values of variables are all invalid, because the line has not been run.
100 %
No issues found
Autos
Search (Ctrl+E)
Search Depth:
Name
Value
Type
a
-858993460
int
b
-858993460
int
max
-858993460
int
Call Stack
Name
testDebug
[External
kernel32.d

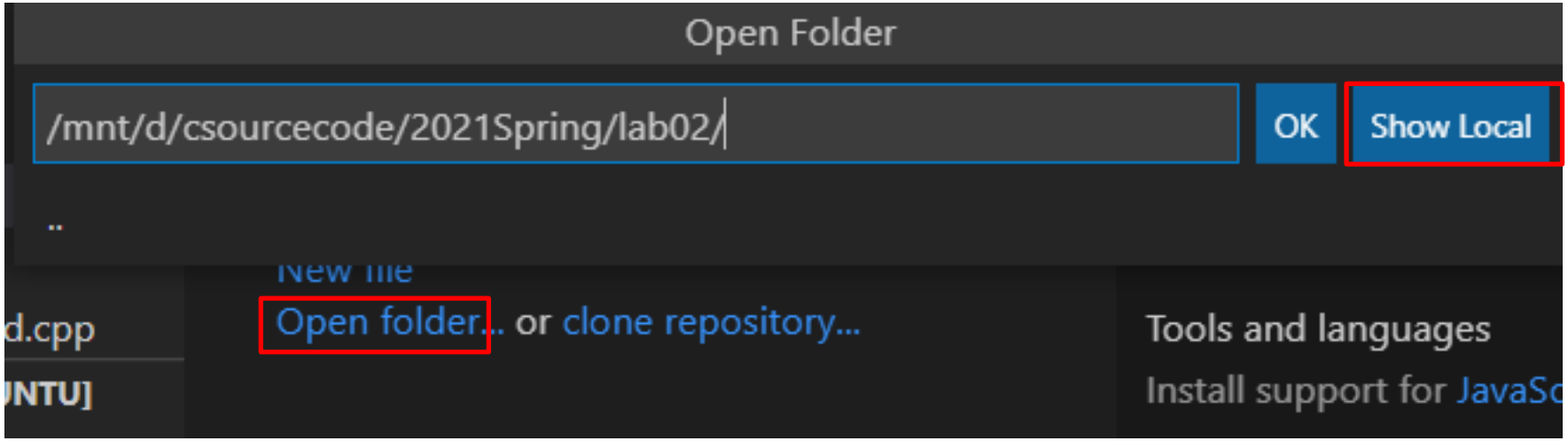

File Edit View Project Build Debug Test Analyze Tools Extensions Window
Windows
Graphics
Continue F5
Break All Ctrl+Alt+Break
Stop Debugging Shift+F5
Detach All
Terminate All
Restart Ctrl+Shift+F5
Apply Code Changes Alt+F10
Performance Profiler... Alt+F2
Relaunch Performance Profiler Shift+Alt+F2
Attach to Process... Ctrl+Alt+P
Other Debug Targets
Step Into F11
Step Over F10
Step Out Shift+F11
QuickWatch... Shift+F9
Toggle Breakpoint F9
New Breakpoint
Delete All Breakpoints Ctrl+Shift+F9
Disable All Breakpoints
Clear All DataTips
Export DataTips ...
Import DataTips ...
Save Dump As...
Options...
testDebug Debug Properties
Process: [12364] testDebug.exe
test.cpp
testDebug
#include <iostream>
using namespace std;
int main()
{
int a, b, max;
a = 10;
b = -4;
if (a > b)
max = a;
else
max = b;
cout << "The max is "
100 %
No issues found
Autos
Search (Ctrl+E)
Search Depth: 3
Name
Value
Type
a 10 int
b -4 int
max -858993460 int
You can also use the menu item in “Debug”menu to run you program .
The line at the yellow arrow will be run for the next.
The value of max is invalid because the assignment has not been run.

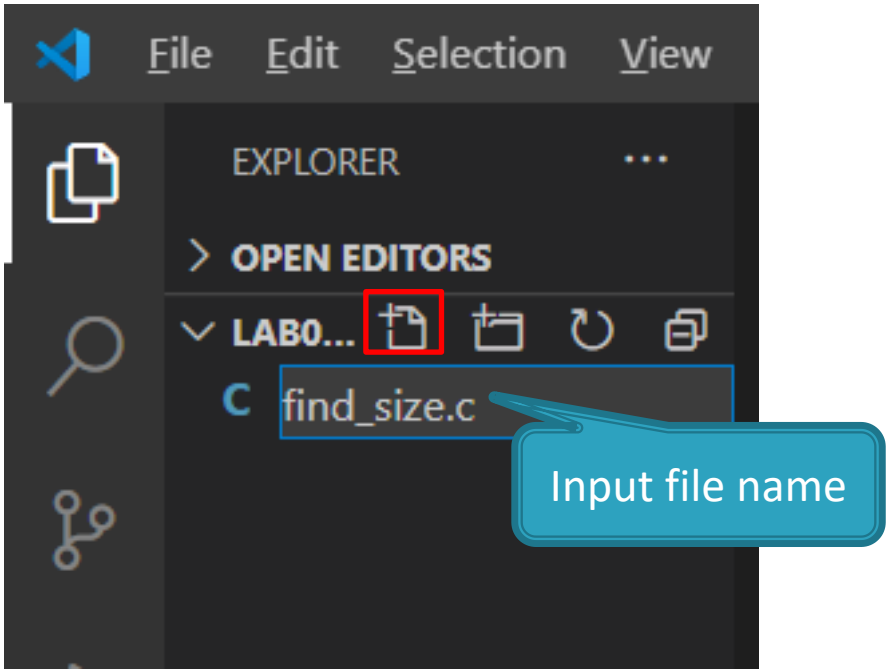

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Debug
x86
Continue
Process: [12364] testDebug.exe
Lifecycle Events
Thread: [13800] Main Thread
test.cpp
testDebug
(Global Scope)
#include <iostream>
using namespace std;
int main()
{
int a, b, max;
a = 10;
if (a > b)
max = a;
else
max = b;
cout << "The max is " << max << endl;
≤1ms elapsed
return 0;
}
If you want to stop debugging, press “stop” button.
Click the breakpoint for the second time can cancel the breakpoint.
100 %
No issues found
Autos
Search (Ctrl+E)
Search Depth: 3
Name
Value
Type
b
-4
int
max
10
int

2.2 Data types



Example: Write a C program to find Size of fundamental data types.
(All examples are written in VScode and compiled under WSL)





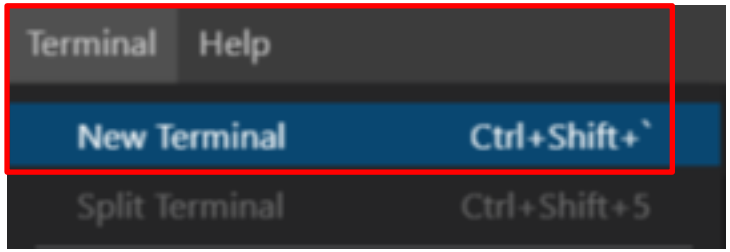
Type the codes as follows:

```c
#include <stdio.h>
#include <stdbool.h>

int main()
{
    printf("\n\nFind Size of the fundamental data t...\n");
    printf("-----------------------------------------\n");
    printf("The sizeof(char) is:        %ld bytes\n", sizeof(char));
    printf("The sizeof(short) is:       %ld bytes\n", sizeof(short));
    printf("The sizeof(int) is:         %ld bytes\n", sizeof(int));
    printf("The sizeof(long) is:        %ld bytes\n", sizeof(long));
    printf("The sizeof(long long) is:   %ld bytes\n", sizeof(long long));
    printf("The sizeof(float) is:       %ld bytes\n", sizeof(float));
    printf("The sizeof(double) is:      %ld bytes\n", sizeof(double));
    printf("The sizeof(long double) is: %ld bytes\n", sizeof(long double));
    printf("The sizeof(bool) is:        %ld bytes\n", sizeof(bool));

    return 0;
}
```

If you use %d, the compiler will give warnings.

```
find_size.c: In function ‘main’:
find_size.c:8:42: warning: format ‘%d’ expects argument of type ‘int’, but argument 2 has type ‘long unsigned int’ [-Wformat=]
    8 |     printf("The sizeof(char) is:       %d bytes\n", sizeof(char));
      |                                          ~^         ~~~~~~~~~~~
      |                                           |         |
      |                                           int       long unsigned int
      |                                          %ld
find_size.c:9:42: warning: format ‘%d’ expects argument of type ‘int’, but argument 2 has type ‘long unsigned int’ [-Wformat=]
    9 |     printf("The sizeof(short) is:      %d bytes\n", sizeof(short));
      |                                          ~^         ~~~~~~~~~~~~
      |                                           |         |
      |                                           int       long unsigned int
      |                                          %ld
```

open the Terminal window to input the commands




```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ gcc find_size.c -o find_size
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ ./find_size

Find Size of the fundamental data types:
-----------------------------------------
The sizeof(char) is:        1 bytes
The sizeof(short) is:       2 bytes
The sizeof(int) is:         4 bytes
The sizeof(long) is:        8 bytes
The sizeof(long long) is:   8 bytes
The sizeof(float) is:       4 bytes
The sizeof(double) is:      8 bytes
The sizeof(long double) is: 16 bytes
The sizeof(bool) is:        1 bytes
```

The output

Example: Write a C++ program to find Size of fundamental data types.

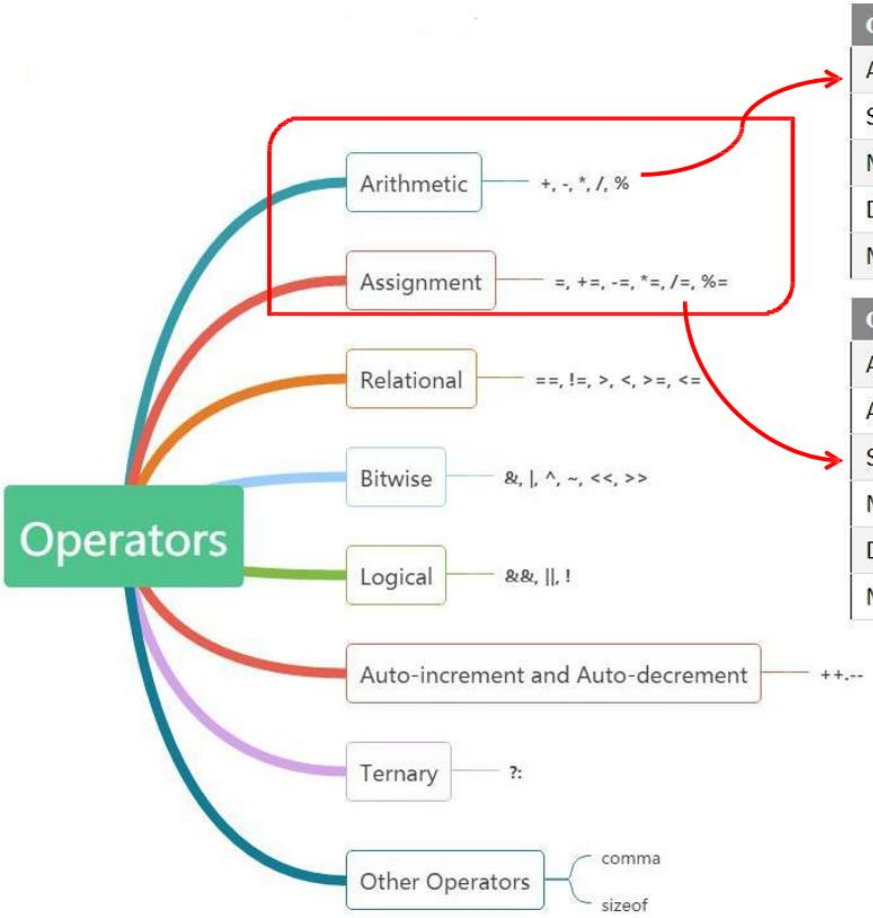
find_size.cpp > ...

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

int main()
{
    cout << "\n\nFind Size of the fundamental data types:\n";
    cout << "-----------------------------------------\n";
    cout << "The size of(char) is:         " << sizeof(char) << endl;
    cout << "The size of(short) is:        " << sizeof(short) << endl;
    cout << "The size of(int) is:          "  << sizeof(int) << endl;
    cout << "The size of(long) is:         " << sizeof(long) << endl;
    cout << "The size of(long long) is:    " << sizeof(long long) << endl;
    cout << "The size of(float) is:        " << sizeof(float) << endl;
    cout << "The size of(double) is:       " << sizeof(double) << endl;
    cout << "The size of(long double) is:  " << sizeof(long double) << endl;
    cout << "The size of(bool) is:         "    << sizeof(bool) << endl;

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ g++ find_size.cpp
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ ls
a.out  find_size  find_size.c  find_size.cpp
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ ./a.out


Find Size of the fundamental data types:
-----------------------------------------
The size of(char) is:         1
The size of(short) is:        2
The size of(int) is:          4
The size of(long) is:         8
The size of(long long) is:    8
The size of(float) is:        4
The size of(double) is:       8
The size of(long double) is:  16
The size of(bool) is:         1
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$
```

2.3 Arithmetic Operators



Operator	Symbol	Form	Operation
Addition	+	x + y	x plus y
Subtraction	-	x - y	x minus y
Multiplication	*	x * y	x multiplied by y
Division	/	x / y	x divided by y
Modulus (Remainder)	%	x % y	The remainder of x divided by y

Operator	Symbol	Form	Operation
Assignment	=	x = y	Assign value y to x
Addition assignment	+=	x += y	Add y to x
Subtraction assignment	-=	x -= y	Subtract y from x
Multiplication assignment	*=	x *= y	Multiply x by y
Division assignment	/=	x /= y	Divide x by y
Modulus assignment	%=	x %= y	Put the remainder of x / y in x

Example Program of Arithmetic Operators:

arithmetic_operators.cpp > ...

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

int main()
{
    //Variable Declaration
    int a = 200;
    int b = 26;
    int c = 50;
    int d = 40;
    int result;

    cout << "Simple Arithmetic Operators Example Program";
    result = a - b;     //substraction
    cout << "\na - b = " << result;

    result = b * c;   //multiplication
    cout << "\nb * c = " << result;

    result = a / c;   //division
    cout << "\na / c = " << result;

    result = a*b + c*d;     //mixed arithmetic operation(precedence)
    cout << "\na*b + c*d = " << result << endl;

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ g++ arithmetic_operators.cpp
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ ./a.out
Simple Arithmetic Operators Example Program
a - b = 174
b * c = 1300
a / c = 4
a*b + c*d = 7200
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$
```

Example of Assignment(Compound Assignment) Operators

assignment_operator.cpp > ...

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

int main()
{
    int num1 = 240;
    int num2 = 40;

    num2 = num1;
    cout << "= Output:" << num2 << endl;

    //compound assignment statements +=,-=,/=,%=
    num2 += num1;     //the same as num2 = num2 + num1;
    cout << "+= Output:" << num2 << endl;
    num2 -= num1;     //the same as num2 = num2 - num1;
    cout << "-= Output:" << num2 << endl;
    num2 *= num1;     //the same as num2 = num2 * num1;
    cout << "*= Output:" << num2 << endl;
    num2 /= num1;     //the same as num2 = num2 / num1;
    cout << "/= Output:" << num2 << endl;
    num2 %= num1;     //the same as num2 = num2 %num1;
    cout << "%= Output:" << num2 << endl;

    return 0;
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ ./a.out
= Output:240
+= Output:480
-= Output:240
*= Output:57600
/= Output:240
%= Output:0
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$
```

2.4 Keyboard input and terminal output

1.Formatting output with ***printf***

printf (*format-control-string, other-arguments*)

format-control-string describes the output format, which consists of conversion specifiers, field widths, precisions and literal characters with percent sign(%).

Conversion specifier	Description
`d`	Display as a *signed decimal integer*.
`i`	Display as a *signed decimal integer*. [*Note:* The `i` and `d` specifiers are *different* when used with `scanf`.]
`o`	Display as an *unsigned octal integer*.
`u`	Display as an *unsigned decimal integer*.
`x` or `X`	Display as an *unsigned hexadecimal integer*. `X` causes the digits `0-9` and the *uppercase* letters `A-F` to be used in the display and `x` causes the digits `0-9` and the *lowercase* letters `a-f` to be used in the display.
`h`, `l` or `ll` (letter "ell")	Place *before* any integer conversion specifier to indicate that a `short`, `long` or `long long` integer is displayed, respectively. These are called length modifiers.
`e` or `E`	Display a floating-point value in *exponential notation*.
`f` or `F`	Display floating-point values in *fixed-point notation* (`F` is supported in the Microsoft Visual C++ compiler in Visual Studio 2015 and higher).
`g` or `G`	Display a floating-point value in either the *floating-point form* `f` or the exponential form `e` (or `E`), based on the magnitude of the value.
`L`	Place before any floating-point conversion specifier to indicate that a `long double` floating-point value should be displayed.

Type	Format Specifier
int	%d
char	%c
float	%f
double	%lf
short int	%hd
unsigned int	%u
long int	%li
long long int	%lli
unsigned long int	%lu
unsigned long long int	%llu
signed char	%c
unsigned char	%c
long double	%Lf

Example:

```
int a=1234;
float f=123.456;
char ch='a';
printf("%8d,%2d\n",a,a);
printf("%f,%8f,%8.1f,%.2f,%.2e\n",f,f,f,f,f);
printf("%3c\n",ch);
```

Sample output:

```
    1234,1234
123.456000,123.456000,    123.5,123.46,1.23e+02
   a
```

2.Reading Formatted input with ***scanf***

scanf (*format-control-string, other-arguments*)

format-control-string describes the formats of input, ***other-arguments*** are **pointers** to variables in which the input will be stored.

Conversion specifier	Description
Integers	
d	Read an *optionally signed decimal integer*. The corresponding argument is a pointer to an `int`.
i	Read an *optionally signed decimal, octal or hexadecimal integer*. The corresponding argument is a pointer to an `int`.
o	Read an *octal integer*. The corresponding argument is a pointer to an unsigned `int`.
u	Read an *unsigned decimal integer*. The corresponding argument is a pointer to an unsigned `int`.
x or X	Read a *hexadecimal integer*. The corresponding argument is a pointer to an unsigned `int`.
h, l and ll	Place *before* any of the integer conversion specifiers to indicate that a `short`, `long` or `long long` integer is to be input, respectively.
Floating-point numbers	
e, E, f, g or G	Read a *floating-point value*. The corresponding argument is a pointer to a floating-point variable.
l or L	Place before any of the floating-point conversion specifiers to indicate that a `double` or `long double` value is to be input. The corresponding argument is a pointer to a `double` or `long double` variable.
Characters and strings	
c	Read a *character*. The corresponding argument is a pointer to a `char`; no null (`'\0'`) is added.
s	Read a *string*. The corresponding argument is a pointer to an array of type `char` that's large enough to hold the string and a terminating null (`'\0'`) character—which is automatically added.

Note: When inputting data, prompt the user for one data item or a few data items at a time. Avoid asking the user to enter many data items in response to a single prompt.

Example:

scandemo.c > ...

```c
#include <stdio.h>

int main()
{
    printf("Please input an integer, a character and a float:\n");
    int a;
    scanf("%d", &a);
    printf("a = %d\n", a);

    getchar();    //discard the newline or space symbol
    char b;
    scanf("%c", &b);
    printf("b = %c\n", b);

    float c;
    scanf("%f", &c);
    printf("c = %f\n", c);

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ gcc scandemo.c
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ ./a.out
Please input an integer, a character and a float:
34  A 56.8
a = 34
b = A
c = 56.799999
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ ./a.out
Please input an integer, a character and a float:
34
a = 34
M
b = M
93.2
c = 93.199997
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$
```

Use blank space key to separate the data

Use Enter key to separate the data

white space, such as space, new line and tab is the valid separator.

3. *cout*

cout << variable1(expression1) [<< variable2 << variable n];

coutdemo.cpp > main()

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

int main()
{
    int a = 10;
    float b = 45.7;
    char c = 'A';

    cout << "a = " << a << ",b = " << b << ",c = " << c << endl;

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ g++ coutdemo.cpp
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ ./a.out
a = 10,b = 45.7,c = A
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$
```

C++ provides two methods to control the **output formats**

- ***Using member functions of ios class***
- ***Using iomanip manipulators***

- ***Using member functions of ios class***

1. cout.setf(fmtflags, fmtflags)

Arguments for `setf(long, long)`

Second Argument	First Argument	Meaning
`ios_base::basefield`	`ios_base::dec`	Use base 10.
	`ios_base::oct`	Use base 8.
	`ios_base::hex`	Use base 16.
`ios_base::floatfield`	`ios_base::fixed`	Use fixed-point notation.
	`ios_base::scientific`	Use scientific notation.
`ios_base::adjustfield`	`ios_base::left`	Use left-justification.
	`ios_base::right`	Use right-justification.
	`ios_base::internal`	Left-justify sign or base prefix, right-justify value.

https://blog.csdn.net/baishuiniyaonulia/article/details/79144033

- ***Using member functions of ios class***

2. cout.width(len) //set the field width
3. cout.fill(ch) // fill character to be used with justified field
4. cout.precision(p) // set the precision of floating-point numbers

coutset.cpp > main()

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

int main()
{
    cout.setf(ios_base::fixed, ios_base::floatfield);
    cout << 56.8;
    cout.width(12);
    cout.fill('+');
    cout << 456.77 << endl;

    cout.precision(2);
    cout << 123.356 << endl;
    cout.precision(5);
    cout << 3897.678485 << endl;

    return 0;


}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ g++ coutset.cpp
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ ./a.out
56.800000++456.770000
123.36
3897.67848
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$
```

- ***Using iomanip manipulators***

#include <iomanip>

1. setw(p)
2. setfill(ch)
3. setprecision(d)

coutmanip.cpp > ...

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

int main()
{
    cout.setf(ios_base::fixed, ios_base::floatfield);
    cout << 56.8 << setw(12) << 456.77 << endl;

    cout << setprecision(2) << 123.356 << endl;
    cout << setprecision(5) << 3897.6784385 << endl;

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ g++ coutmanip.cpp
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ ./a.out
56.800000  456.770000
123.36
3897.67844
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$
```

cout.unsetf( )

coutunset.cpp > ...

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

int main()
{
    cout.setf(ios_base::fixed, ios_base::floatfield);
    cout << 56.8 << setw(12) << 456.77 << endl;

    cout << setprecision(2) << 123.356 << endl;
    cout << setprecision(5) << 3897.6784385 << endl;

    cout << '\n';
    cout.unsetf(ios_base::fixed);
    cout << 56.8 << setw(12) << setfill('#') << 456.77 << endl;

    cout << setprecision(2) << 123.356 << endl;
    cout << setprecision(5) << 3897.678385 << endl;

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ g++ coutunset.cpp
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ ./a.out
56.800000  456.770000
123.36
3897.67844

56.8######456.77
1.2e+02
3897.7
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$
```

4. *cin*

cin >> variable1 [>> variable2 >> …variable n];

cincout.cpp > ...

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

int main()
{
    cout << "Please input an integer, a character and a float\n";
    int a;
    cin >> a;
    cout << "a = " << a << endl;

    char b;
    cin >> b;
    cout << "b = " << b << endl;

    float c;
    cin >> c;
    cout << "c = " << c << endl;

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ g++ cincout.cpp
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ ./a.out
Please input an integer, a character and a float
5 G 4.9
a = 5
b = G
c = 4.9
maydlee@LAPTOP-U1MO0N2F:/mnt/d/csourcecode/2021Spring/lab02$ ./a.out
Please input an integer, a character and a float
43 Y 87.4
a = 43
b = Y
c = 87.4
```

Use blank space key to separate the data

Use Enter key to separate the data

white space, such as space, new line and tab is the valid separator.

3 Exercises

1. Write a program to produce the output as shown below.

```
Result:
x value y value Expressions      Result
10 |     5 |      x=y+3          |x=8
10 |     5 |      x=y-2          |x=3
10 |     5 |      x=y*5          |x=25
10 |     5 |      x=x/y          |x=2
10 |     5 |      x=x%y          |x=0
```

2. Write a program that asks the user to enter the number of seconds as an integer value(use type long, or, if available, long long) and then displays the equivalent time in days, hours, minutes and seconds. Use symbolic constants to represent the number of hours in the day, the number of minutes in an hour, and the number of seconds in a minute. The output should look like this:

```
Enter the number of seconds:31600000
31600000 seconds = 365 days, 17 hours, 46 minutes, 40 seconds
```

3. Write a .**C** program that asks the user to enter an integer value, a character, and a float value. And then use the **printf** statement to print them out. A sample run should look like this:

```
Please Enter a Character :  A
Please Enter an Integer Value :  20
Please Enter Float Value :  30.678

The variables you entered were:
The Character Value that you Entered is :  A
The Integer Value that you Entered is : 20
The Float Value that you Entered is :  30.678
```

What happens when you are prompted to enter an integer, but you enter a float?