

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/>

The screenshot shows the Visual Studio download page in Chinese. The main heading is "Visual Studio 面向任何开发者的同类最佳工具". Below this, there are three columns for different versions: Visual Studio, Visual Studio Code, and Visual Studio for Mac. The "Visual Studio" column has a red box around the "Community 2019" download button. A blue callout bubble points to this button with the text "Click Visual Studio Community 2019". The page also features a "请帮我选择!" (Help me choose!) section with a form to describe the user's project and development goals. The browser's address bar shows the URL "https://visualstudio.microsoft.com/zh-hans/". The Windows taskbar at the bottom shows the system clock as 11:51 on 2020/8/28.

Visual Studio
面向任何开发者的同类最佳工具

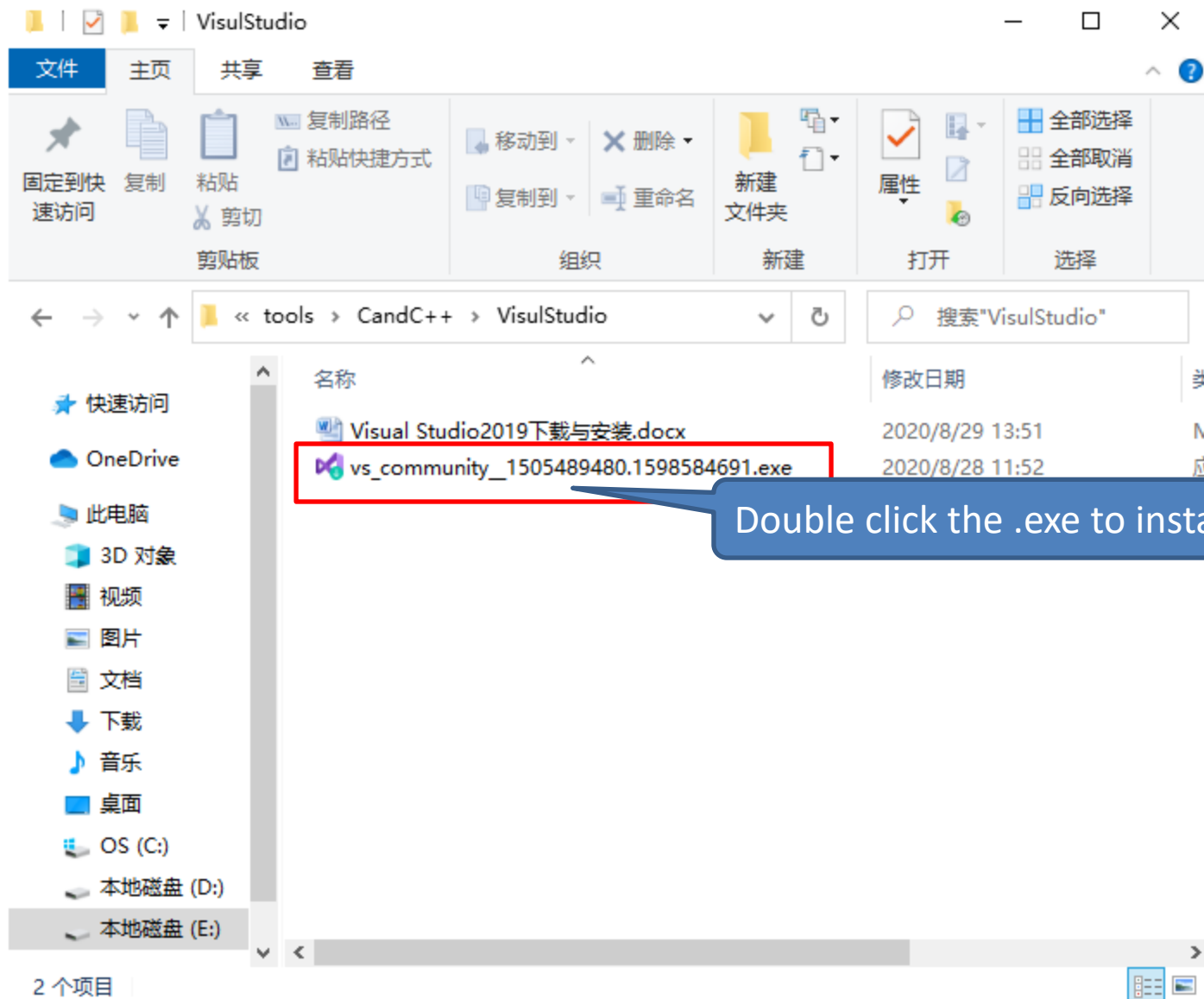
Visual Studio Code
可在任意操作系统上进行编辑和调试
(使用 Visual Studio Code 即表示你同意其 [许可和隐私声明](#))

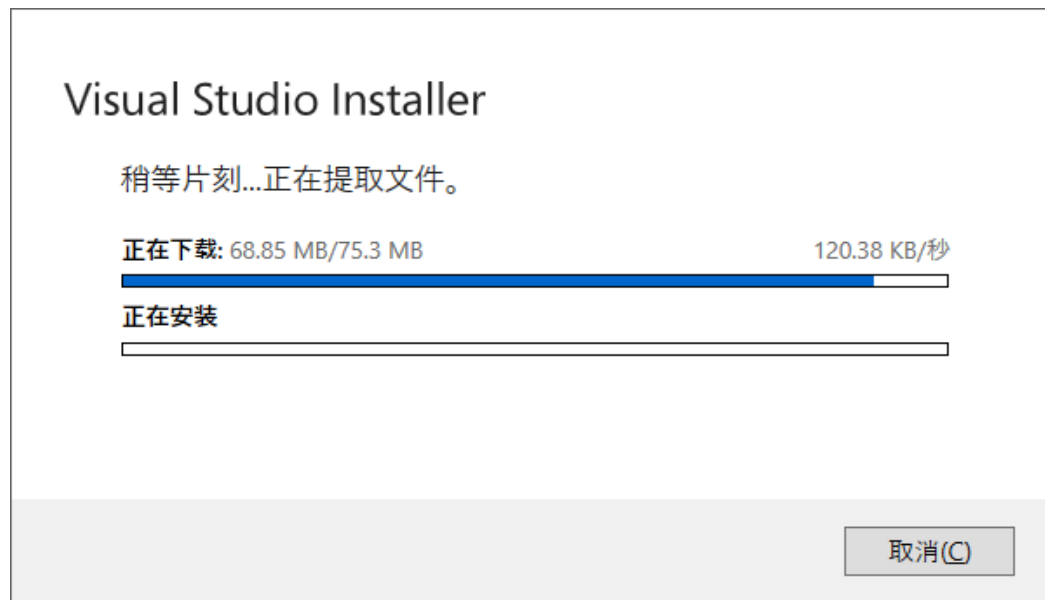
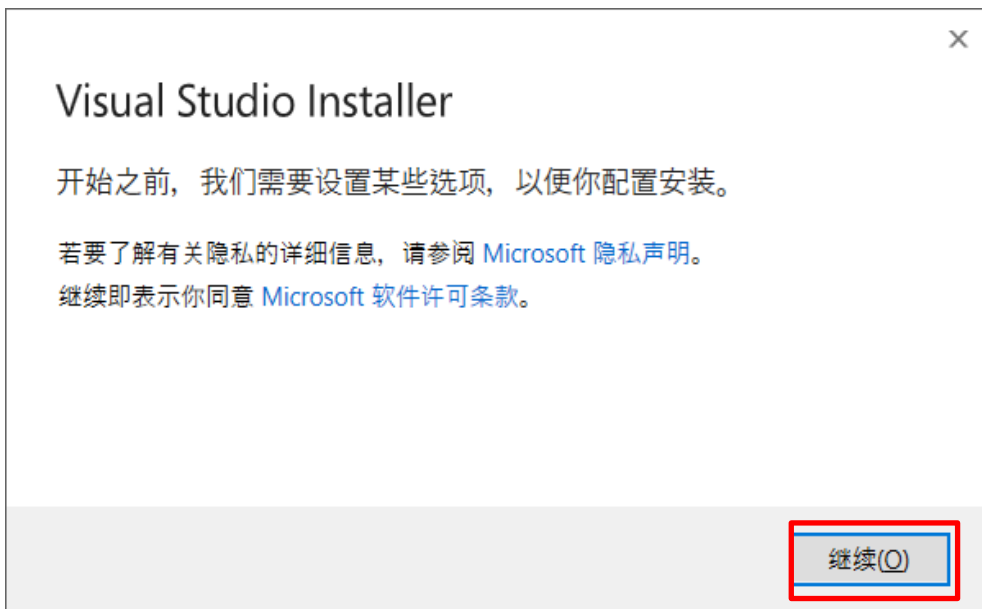
Visual Studio for Mac
使用 .NET 开发 iOS、Android 和 Web 应用和游戏
[阅读有关激活许可证的详细信息](#)

下载 Visual Studio
Community 2019
Professional 2019
Enterprise 2019

请帮我选择!
请描述你的项目,
我的开发目标

Click Visual Studio Community 2019





正在安装 — Visual

Workloads

工作负载

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

使用所选工具(包括 MSVC、Clang、CMake 或 MSBuild)生成适用于 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 iOS 和 Android 平台创建应用程序。

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)

□ 适用于最新 v142 生成工具的 C++ MFC (x86 和 x...

□ 适用于 Windows 的 C++ Clang 工具(10.0.0 - x64...

Choose "Desktop envelopment with C++"

位置

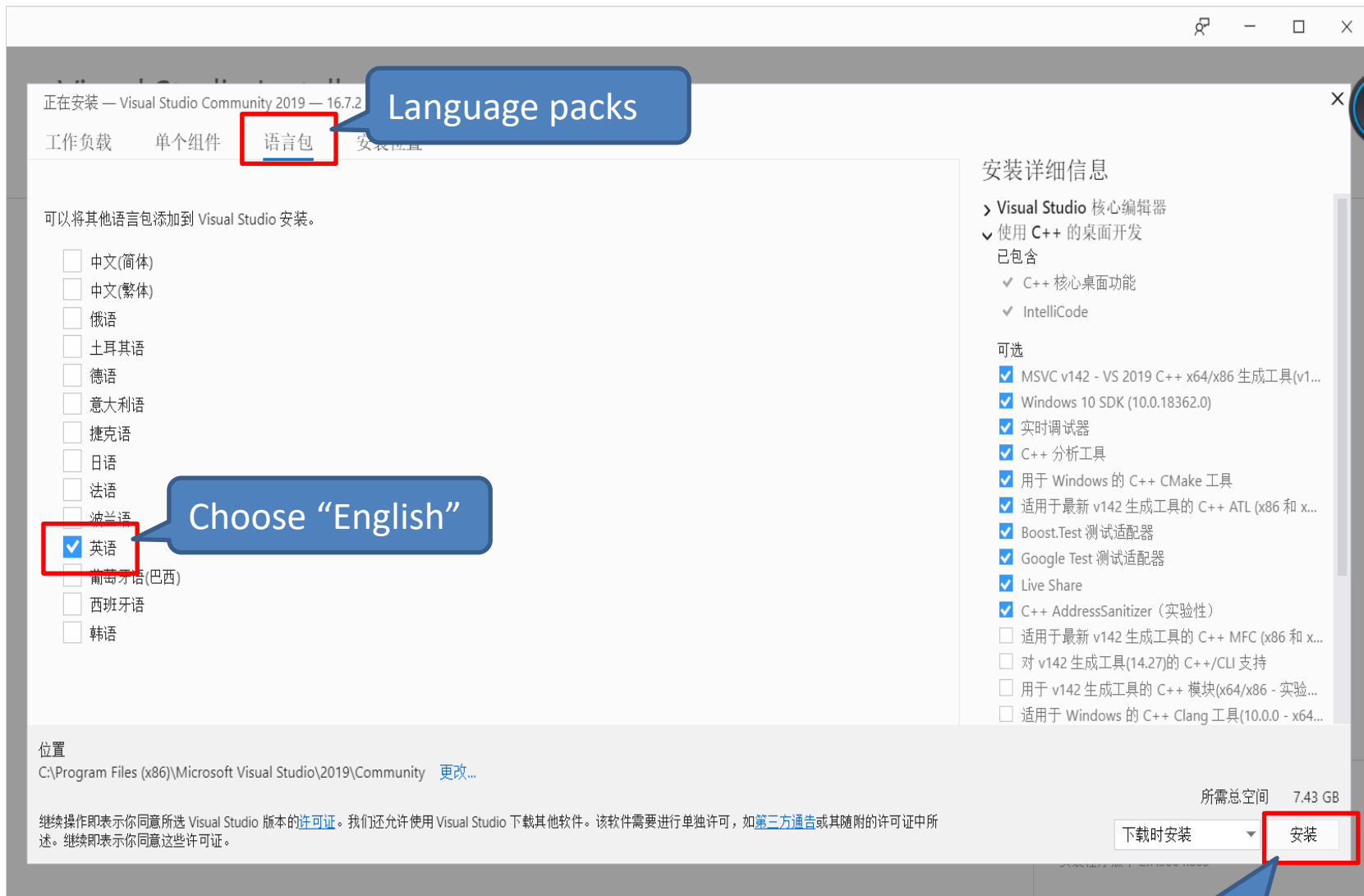
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community 更改...

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。

所需总空间 7.46 GB

下载时安装

安装



Visual Studio Installer

已安装

可用



Visual Studio Community 2019

16.7.2

⚠ 需要重启。若需要，则重启后将自动恢复所有剩余的设置。

重启

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](#)

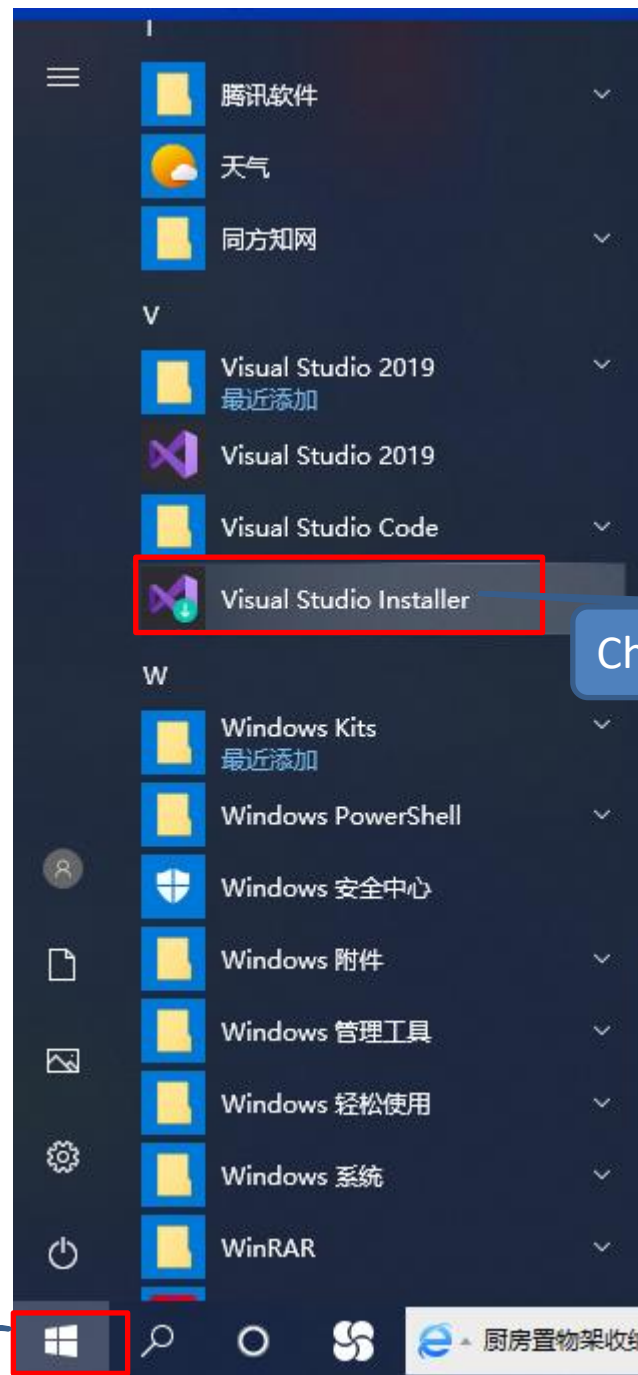
It's an exciting time to be writing code! Especiall...

2020年8月29日星期六

[查看更多联机...](#)

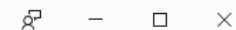
需要帮助? 请参阅 [Microsoft 开发者社区](#) 或通过 [Visual Studio 支持](#) 与我们联系。

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!](#)

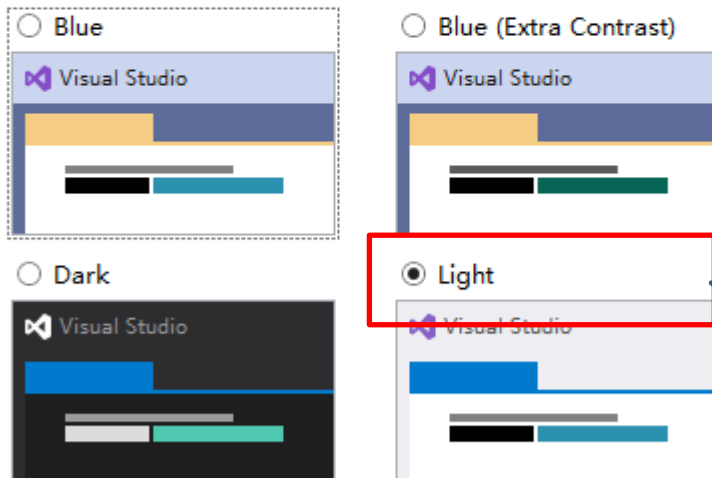
[Not now, maybe later.](#)

Visual Studio

Start with a familiar environment

Development Settings: General

Choose your color theme



X

Choose color theme

Start Visual Studio

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 the 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

Create a new project

Search for templates (Alt+S)

All languages

All platforms

All project types

Recent project templates

A list of your recently accessed templates will be displayed here.



Empty Project

Start from scratch with C++ for Windows. Provides no starting files.

C++

Windows

Console



Console App

Run code in a Windows terminal. Prints "Hello World" by default.

C++

Windows

Console



Windows Desktop Wizard

Create your own Windows app using a wizard.

C++

Windows

Console

Desktop

Library



Windows Desktop Application

A project for an application with a graphical user interface that runs on Windows.

C++

Windows

Desktop

Choose "Console App"

Back

Next

Click "Next" button

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

The screenshot displays the Microsoft Visual Studio IDE with a C++ project named 'HelloWorld'. The main window shows the source code for 'HelloWorld.cpp', which includes a simple 'main' function that prints 'Hello World!'. The code is annotated with a red box and a callout stating 'These simple codes are written automatically'. The 'Local Windows Debugger' button in the toolbar is highlighted with a red box and a callout 'Run your program'. The Solution Explorer on the right shows the project structure, with 'HelloWorld' selected and highlighted by a red box and a callout 'Project framework'. The Debug Console at the bottom shows the output 'Hello World!' highlighted with a red box and a callout 'Output of your program'. Other callouts include 'Some guides of usage' pointing to the code comments and 'These simple codes are written automatically' pointing to the code itself.

```
1 // HelloWorld.cpp : This file contains the 'main' function. Program execution begins and ends there.
2 //
3
4 #include <iostream>
5
6 int main()
7 {
8     std::cout << "Hello World!\n";
9 }
10
11 // Run program: Ctrl + F5 or Debug > Start Without Debugging menu
12 // Debug program: F5 or Debug > Start Debugging menu
13
14 // Tips for Getting Started:
15 // 1. Use the Solution Explorer window to add/manage files
16 // 2. Use the Team Explorer window to connect to source control
17 // 3. Use the Output window to see build output and other messages
18 // 4. Use the Error List window to view errors
19 // 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
20 // 6. In the future, to open this project again, go to File > Open > Project and select the .sln file
21
```

Microsoft Visual Studio
Hello World!

D:\sourcecode\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 . . .

Create a an Empty Project

Create a new project

Recent project templates

Search for templates (Alt+S)

All languages

All platforms

All project types



Empty Project

Start from scratch with C++ for Windows. Provides no starting files.

C++

Windows

Console



Console App

Run code in a Windows terminal. Prints "Hello World" by default.

C++

Windows

Console



Windows Desktop Wizard

Create your own Windows app using a wizard.

C++

Windows

Console

Desktop

Library



Windows Desktop Application

A project for an application with a graphical user interface that runs on Windows.

C++

Windows

Desktop

Back

Next

Create an "Empty Project"

Click "Next" button

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

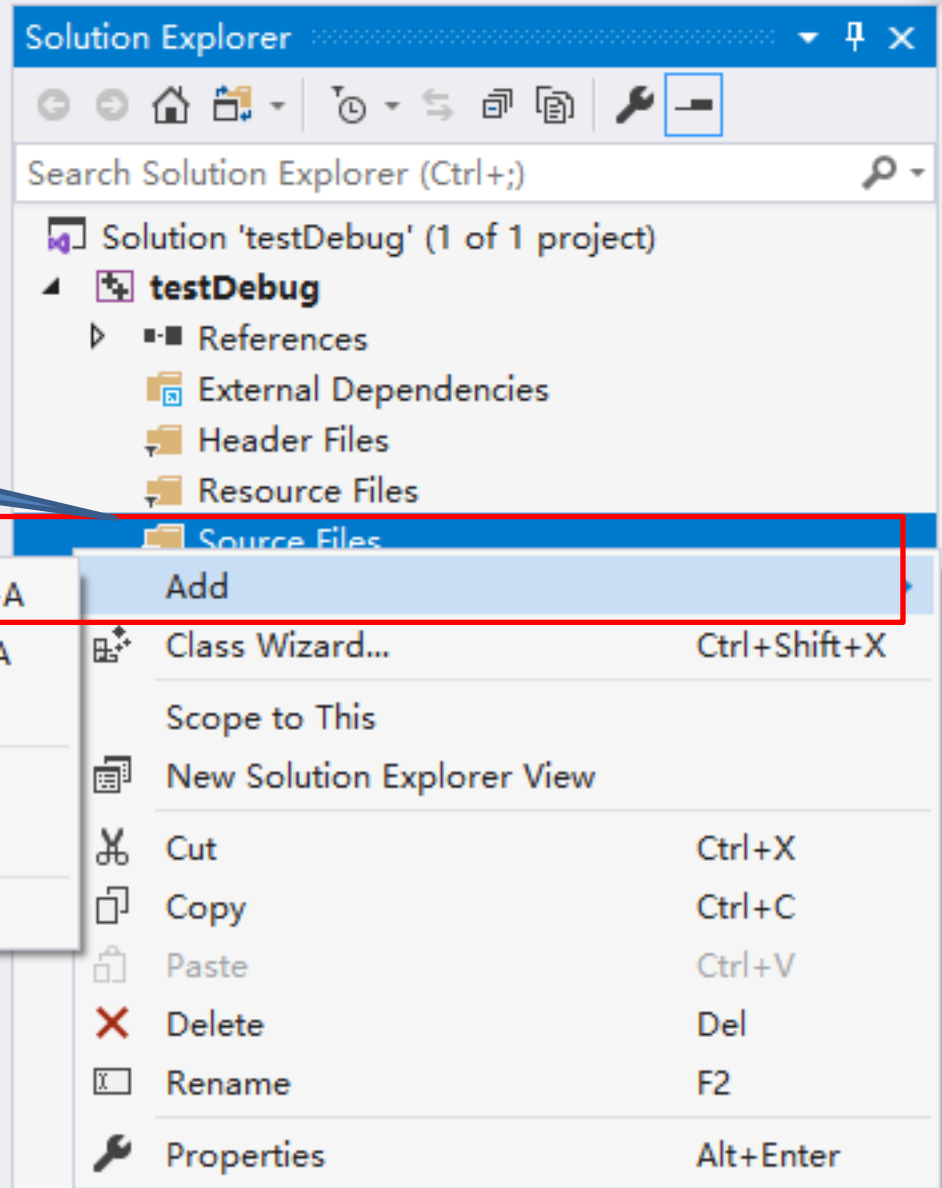
☐ Place solution and project in the same directory

Click "Create" button

Back

Create

Right click mouse on “Source Files” folder, choose
“Add”→“New Item...”



Add New Item - testDebug

Installed

Visual C++

Code
Formatting
ATL
Data
Resource
Web
Utility
Property Sheets
Test
HLSL
Graphics

Online

Sort by: Default



C++ File (.cpp)

Visual C++



Header File (.h)

Visual C++



C++ Class

Visual C++

Choose "C++ File(.cpp)"

Search (Ctrl+E)

Type: Visual C++

Creates a file containing C++ source code

Name:

test.cpp

Location:

D:\sourcecode\testDebug\testDebug\

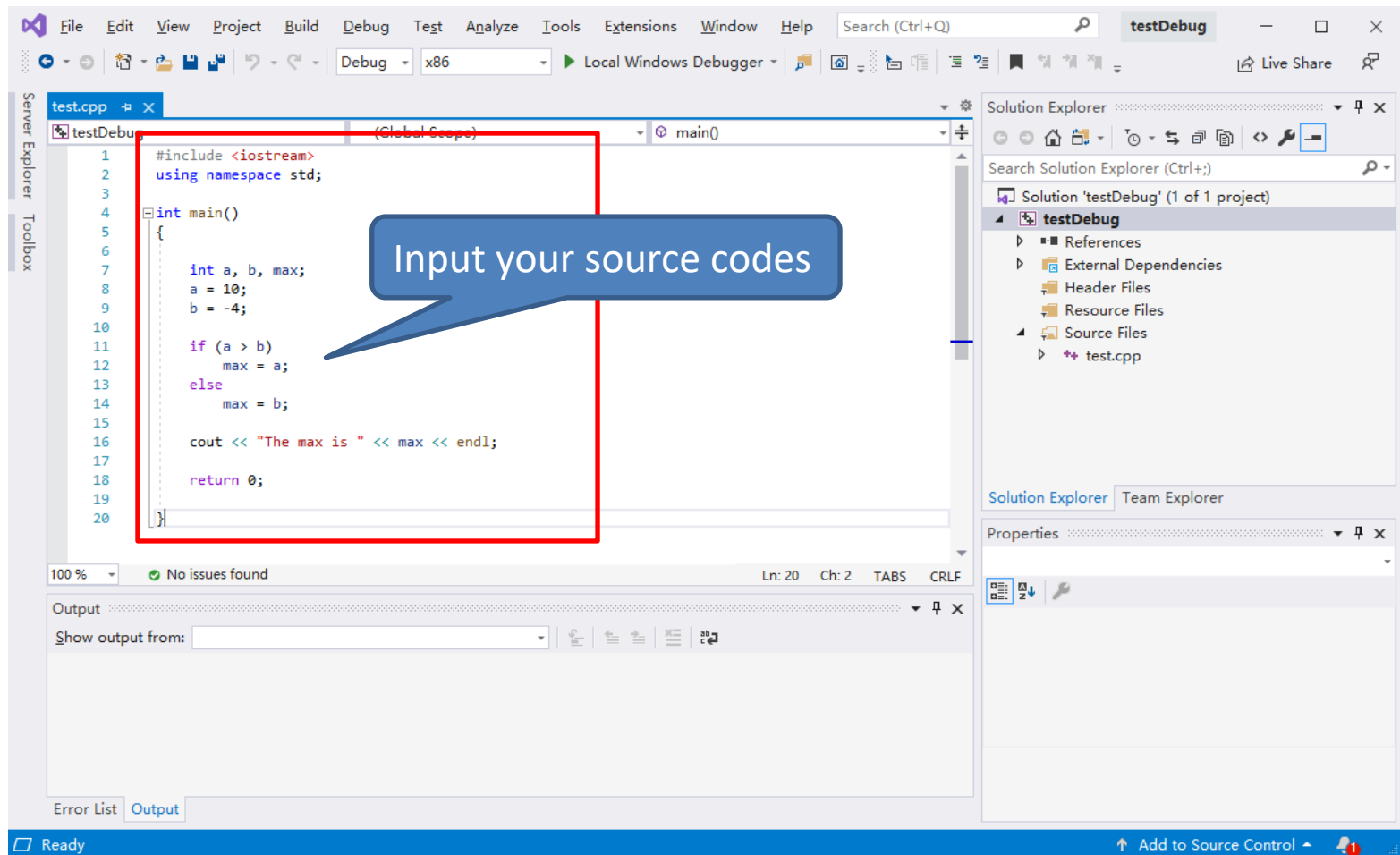
Browse...

Add

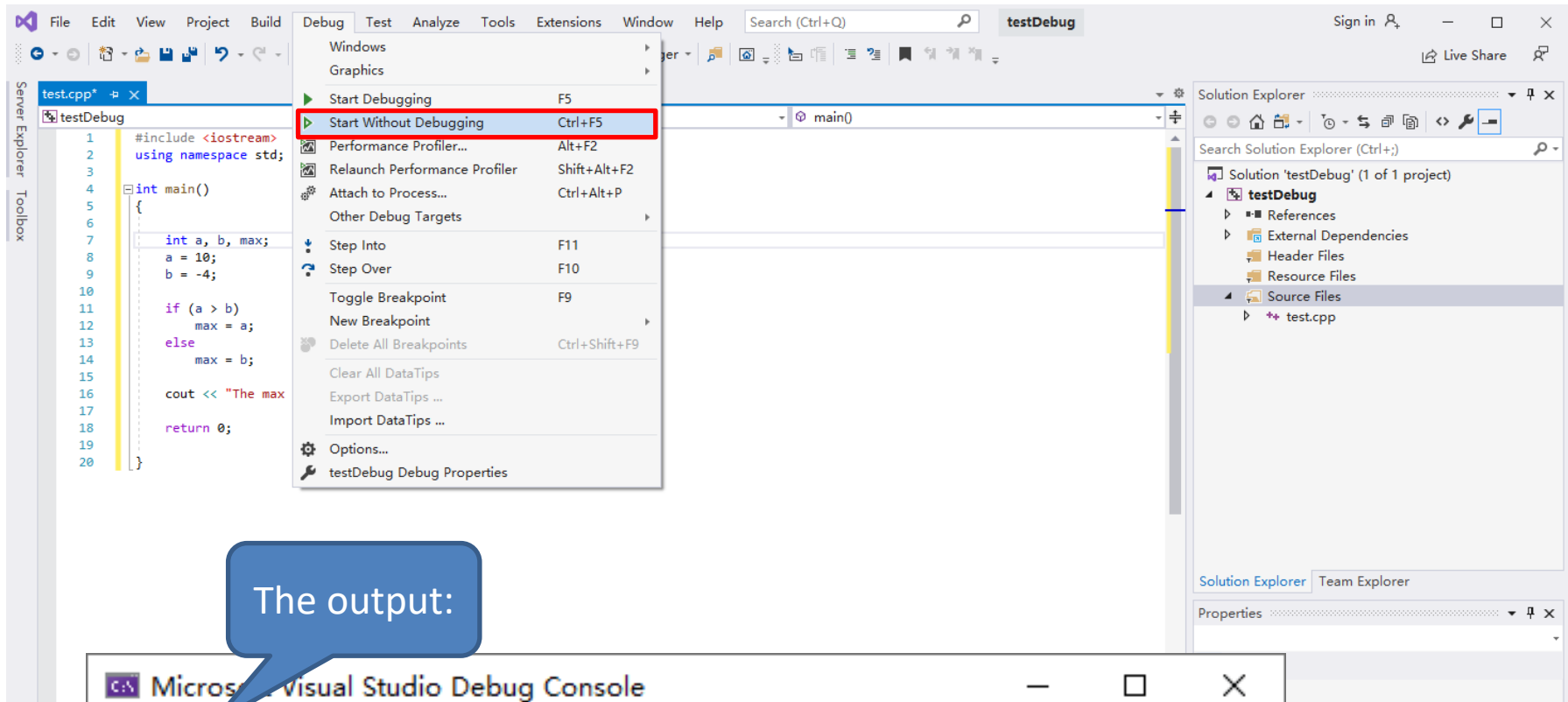
Cancel

Click "Add" button

Input your source name, don't modify the extension name



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



The output:

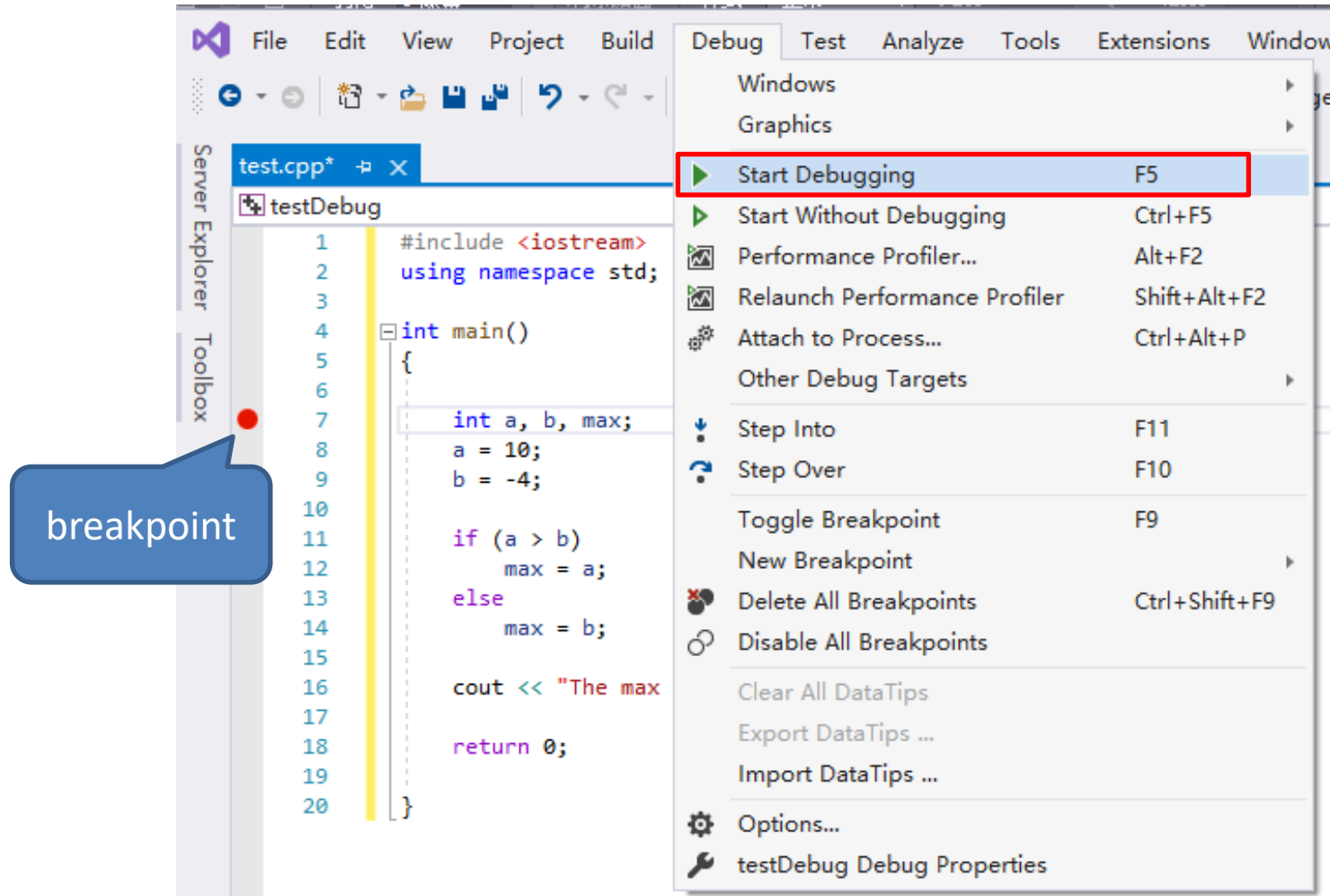
A screenshot of the Visual Studio Debug Console window. The output text is:

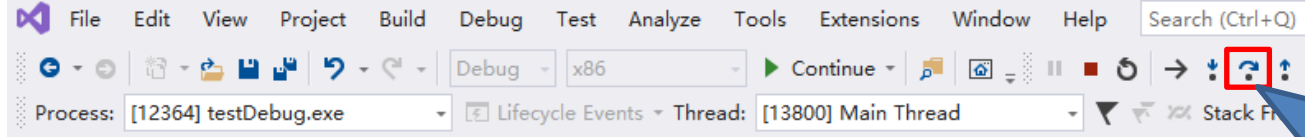
```
C:\> Microsoft Visual Studio Debug Console
The max is 10
D:\sourcecode\testDebug\Debug\testDebug.exe (process 1220) exited
with code 0.
Press any key to close this window . . .
```

The first line, 'The max is 10', is highlighted with a red rectangle.

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.





③ Press “step over” button (or F10) to run the program step by step.

① The program stops at the breakpoint.

```
test.cpp x
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 }
```

② The 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 C
kernel32.d

File Edit View Project Build Debug Test Analyze Tools Extensions Window

Process: [12364] testDebug.exe

test.cpp

```
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 "
```

Debug menu options:

- 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

Autos window:

Name	Value	Type
a	10	int
b	-4	int
max	-858993460	int

The line at the yellow arrow will be run for the next.

You can also use the menu item in "Debug" menu to run your program.

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)

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6
7     int a, b, max;
8     a = 10;
9
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 }
```

Click the breakpoint for the second time can cancel the breakpoint.

If you want to stop debugging, press "stop" button.

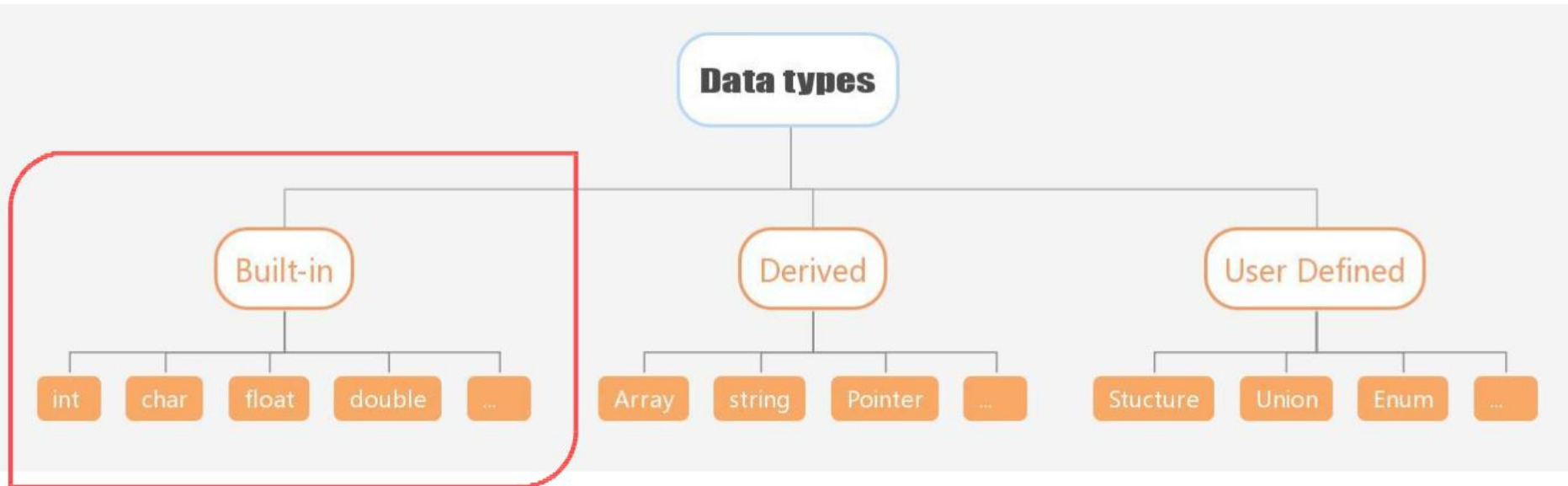
100 % No issues found

Autos

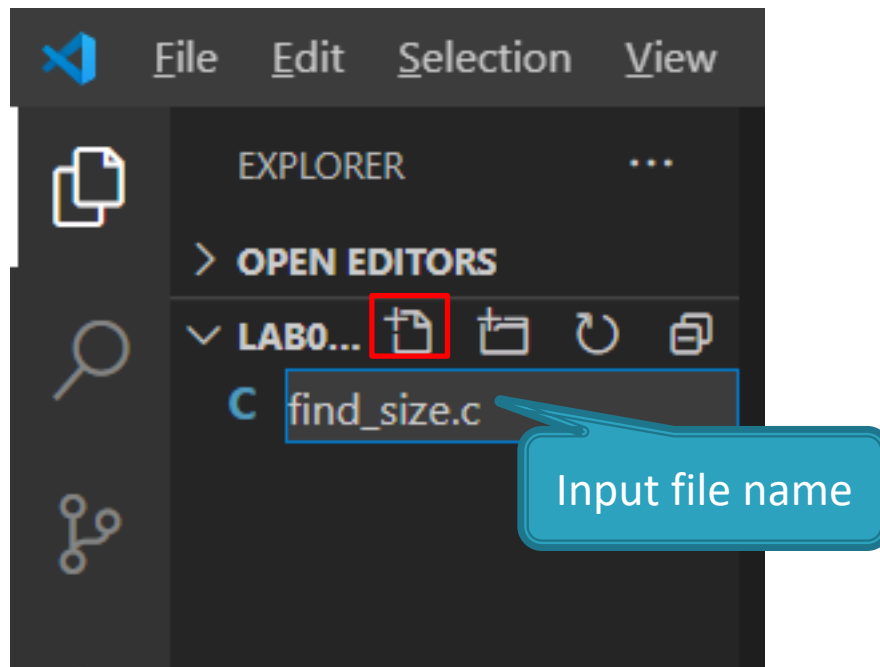
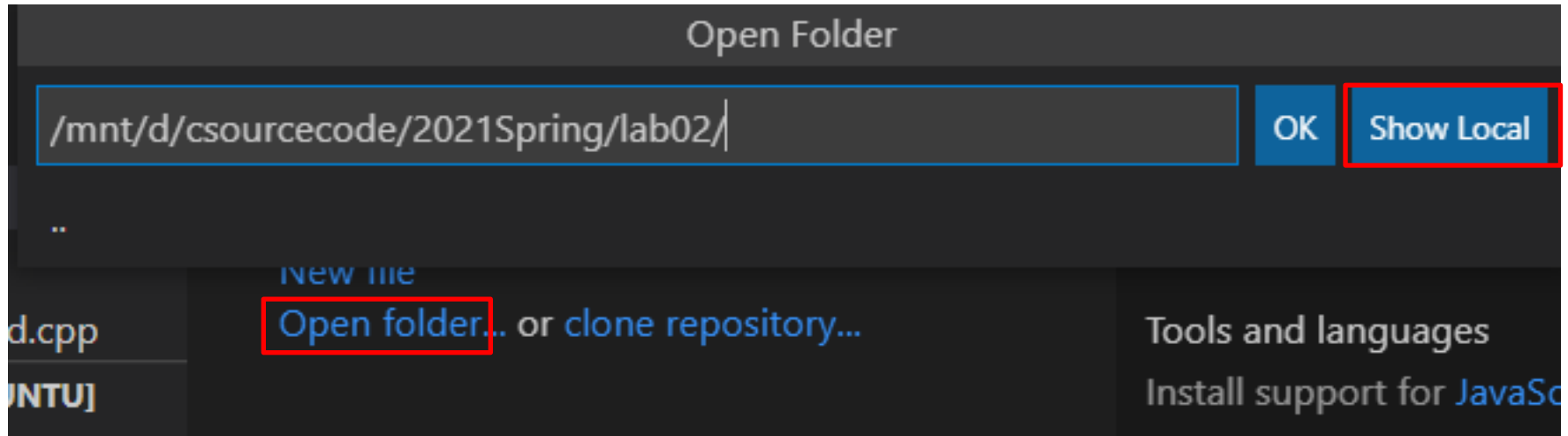
Search (Ctrl+E) Search Depth: 3

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:

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

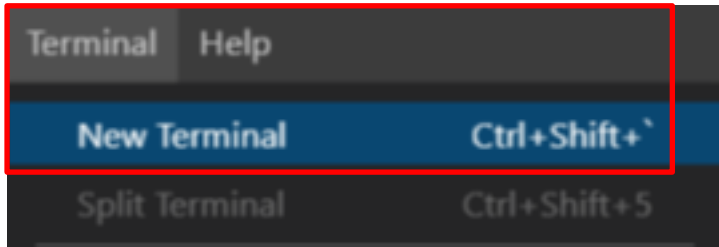
```
#include <stdio.h>
#include <stdbool.h>

int main()
{
    printf("\n\nFind Size of the fundamental data types.\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;
}
```

```
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.

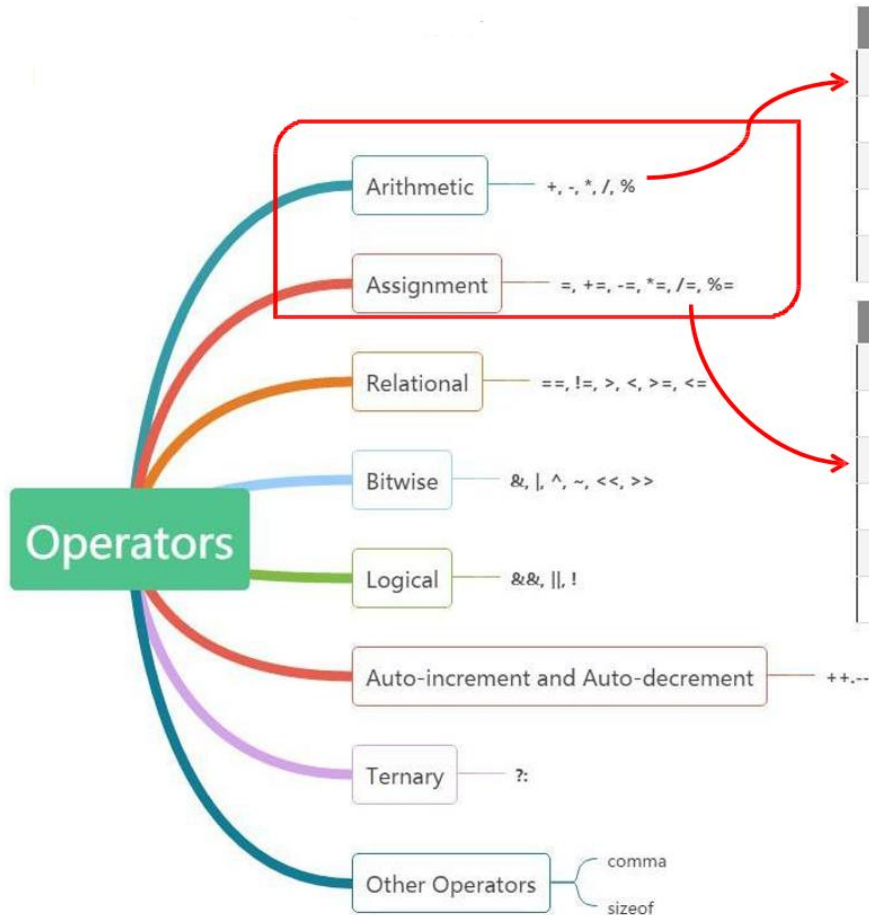
```
g++ find_size.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << "\n\nFind Size of the fundamental data types:\n";
7      cout << "-----\n";
8      cout << "The size of(char) is:          " << sizeof(char) << endl;
9      cout << "The size of(short) is:         " << sizeof(short) << endl;
10     cout << "The size of(int) is:            " << sizeof(int) << endl;
11     cout << "The size of(long) is:           " << sizeof(long) << endl;
12     cout << "The size of(long long) is:       " << sizeof(long long) << endl;
13     cout << "The size of(float) is:          " << sizeof(float) << endl;
14     cout << "The size of(double) is:         " << sizeof(double) << endl;
15     cout << "The size of(long double) is:    " << sizeof(long double) << endl;
16     cout << "The size of(bool) is:          " << sizeof(bool) << endl;
17
18     return 0;
19 }
20
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab02$ g++ find_size.cpp
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab02$ ls
a.out find_size find_size.c find_size.cpp
maydlee@LAPTOP-U1M00N2F:/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-U1M00N2F:/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 > ...
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      //Variable Declaration
7      int a = 200;
8      int b = 26;
9      int c = 50;
10     int d = 40;
11     int result;
12
13     cout << "Simple Arithmetic Operators Example Program";
14     result = a - b;    //subtraction
15     cout << "\na - b = " << result;
16
17     result = b * c;    //multiplication
18     cout << "\nb * c = " << result;
19
20     result = a / c;    //division
21     cout << "\na / c = " << result;
22
23     result = a*b + c*d;    //mixed arithmetic operation(precedence)
24     cout << "\na*b + c*d = " << result << endl;
25
26     return 0;
27 }
28
29
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab02$ g++ arithmetic_operators.cpp
maydlee@LAPTOP-U1M00N2F:/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-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab02$
```

Example of Assignment(Compound Assignment) Operators

```
assignment_operator.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int num1 = 240;
7      int num2 = 40;
8
9      num2 = num1;
10     cout << "= Output:" << num2 << endl;
11
12     //compound assignment statements +=, -=, /=, %=
13     num2 += num1;    //the same as num2 = num2 + num1;
14     cout << "+= Output:" << num2 << endl;
15     num2 -= num1;    //the same as num2 = num2 - num1;
16     cout << "-= Output:" << num2 << endl;
17     num2 *= num1;    //the same as num2 = num2 * num1;
18     cout << "*= Output:" << num2 << endl;
19     num2 /= num1;    //the same as num2 = num2 / num1;
20     cout << "/= Output:" << num2 << endl;
21     num2 %= num1;    //the same as num2 = num2 %num1;
22     cout << "%= Output:" << num2 << endl;
23
24     return 0;
25 }
26
27
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab02$ ./a.out
= Output:240
+= Output:480
-= Output:240
*= Output:57600
/= Output:240
%= Output:0
maydlee@LAPTOP-U1M00N2F:/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 <i>signed decimal integer</i> .
i	Display as a <i>signed decimal integer</i> . [Note: The i and d specifiers are <i>different</i> when used with scanf.]
o	Display as an <i>unsigned octal integer</i> .
u	Display as an <i>unsigned decimal integer</i> .
x or X	Display as an <i>unsigned hexadecimal integer</i> . X causes the digits 0–9 and the <i>uppercase</i> letters A–F to be used in the display and x causes the digits 0–9 and the <i>lowercase</i> letters a–f to be used in the display.
h, l or ll (letter “ell”)	Place <i>before</i> 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 <i>exponential notation</i> .
f or F	Display floating-point values in <i>fixed-point notation</i> (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 <i>floating-point form</i> 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; 默认右对齐

float f=123.456;

char ch='a'; 跨过域宽

printf("%08d,%02d\n",a,a);

printf("%f,%8f,%8.1f,%.2f,%.2e\n",f,f,f,f,f);

printf("%03c\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
<i>Integers</i>	
d	Read an <i>optionally signed decimal integer</i> . The corresponding argument is a pointer to an <code>int</code> .
i	Read an <i>optionally signed decimal, octal or hexadecimal integer</i> . The corresponding argument is a pointer to an <code>int</code> .
o	Read an <i>octal integer</i> . The corresponding argument is a pointer to an unsigned <code>int</code> .
u	Read an <i>unsigned decimal integer</i> . The corresponding argument is a pointer to an unsigned <code>int</code> .
x or X	Read a <i>hexadecimal integer</i> . The corresponding argument is a pointer to an unsigned <code>int</code> .
h, l and ll	Place <i>before</i> any of the integer conversion specifiers to indicate that a short, long or long long integer is to be input, respectively.
<i>Floating-point numbers</i>	
e, E, f, g or G	Read a <i>floating-point value</i> . 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.
<i>Characters and strings</i>	
c	Read a <i>character</i> . The corresponding argument is a pointer to a <code>char</code> ; no null (<code>'\0'</code>) is added.
s	Read a <i>string</i> . The corresponding argument is a pointer to an array of type <code>char</code> that's large enough to hold the string and a terminating null (<code>'\0'</code>) 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 > ...
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("Please input an integer, a character and a float:\n");
6      int a;
7      scanf("%d", &a);
8      printf("a = %d\n", a);
9
10     getchar(); //discard the newline or space symbol
11     char b;
12     scanf("%c", &b);
13     printf("b = %c\n", b);
14
15     float c;
16     scanf("%f", &c);
17     printf("c = %f\n", c);
18
19     return 0;
20 }
21
```

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab02$ gcc scandemo.c
maydlee@LAPTOP-U1M00N2F:/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-U1M00N2F:/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-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab02$
```

Use blank space key to separate the data

Use Enter key to separate the data

3. *cout*

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

```
coutdemo.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a = 10;
7      float b = 45.7;
8      char c = 'A';
9
10     cout << "a = " << a << ",b = " << b << ",c = " << c << endl;
11
12     return 0;
13 }
14
```

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
<code>ios_base::basefield</code>	<code>ios_base::dec</code>	Use base 10.
	<code>ios_base::oct</code>	Use base 8.
	<code>ios_base::hex</code>	Use base 16.
<code>ios_base::floatfield</code>	<code>ios_base::fixed</code>	Use fixed-point notation.
	<code>ios_base::scientific</code>	Use scientific notation.
<code>ios_base::adjustfield</code>	<code>ios_base::left</code>	Use left-justification.
	<code>ios_base::right</code>	Use right-justification.
	<code>ios_base::internal</code>	Left-justify sign or base prefix, right-justify value.

● *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()
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout.setf(ios_base::fixed, ios_base::floatfield);
7      cout << 56.8;
8      cout.width(12);
9      cout.fill('+');
10     cout << 456.77 << endl;
11
12     cout.precision(2);
13     cout << 123.356 << endl;
14     cout.precision(5);
15     cout << 3897.678485 << endl;
16
17     return 0;
18
19
20 }
```

作用域只有一个

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab02$ g++ coutset.cpp
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab02$ ./a.out
56.800000++456.770000
123.36
3897.67848
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab02$
```

● *Using iomanip manipulators*

#include <iomanip>

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

```
coutmanip.cpp > ...
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  int main()
6  {
7      cout.setf(ios base::fixed, ios base::floatfield);
8      cout << 56.8 << setw(12) << 456.77 << endl;
9
10     cout << setprecision(2) << 123.356 << endl;
11     cout << setprecision(5) << 3897.6784385 << endl;
12
13     return 0;
14 }
15
```

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 > ...
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4
5  int main()
6  {
7      cout.setf(ios_base::fixed, ios_base::floatfield);
8      cout << 56.8 << setw(12) << 456.77 << endl;
9
10     cout << setprecision(2) << 123.356 << endl;
11     cout << setprecision(5) << 3897.6784385 << endl;
12
13     cout << '\n';
14     cout.unsetf(ios_base::fixed);
15     cout << 56.8 << setw(12) << setfill('#') << 456.77 << endl;
16
17     cout << setprecision(2) << 123.356 << endl;
18     cout << setprecision(5) << 3897.678385 << endl;
19
20     return 0;
21 }
22
```

科学计数法是有有效位数

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab02$ g++ coutunset.cpp
```

```
maydlee@LAPTOP-U1M00N2F:/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-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab02$
```

4. *cin*

`cin >> variable1 [>> variable2 >> ...variable n];`

```
cincout.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << "Please input an integer, a character and a float\n";
7      int a;
8      cin >> a;
9      cout << "a = " << a << endl;
10
11     char b;
12     cin >> b;
13     cout << "b = " << b << endl;
14
15     float c;
16     cin >> c;
17     cout << "c = " << c << endl;
18
19     return 0;
20 }
21
```

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
maydlee@LAPTOP-U1M00N2F:/mnt/d/csourcecode/2021Spring/lab02$ g++ cincout.cpp
maydlee@LAPTOP-U1M00N2F:/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
```

Use blank space key to separate the data

```
maydlee@LAPTOP-U1M00N2F:/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 Enter key to separate the data