

Introduction to Computer and Programming

Lab course

2023.09.12

Outline

- TA Course Schedule
- Exercise Demo Rules
- Homework Submission Rules
- Introduce the C language and Print function
- Exercise

TA Course Schedule

Date	Content of the Course
9/12	Introduce the C language and Print function
9/19	Logic Operator and Selection
9/26	Loops
10/3	Array and String
10/10	no class
10/17	File I/O
10/24	Functions
10/31	Midterm exam
11/7	Recursive Function
11/14	Pointer
11/21	Pointer 2
11/28	Structure
12/5	Multiple files for program project
12/12	Sorting
12/19	Final exam
12/26	optional
1/2	Make-up Exam(optional)

TA Groups

- TA office: EC118 & EC210
- TA Contact

TA name	email
顏廷恩	tny0302.en12@nycu.edu.tw
吳秉宸	evan20010126.cs12@nycu.edu.tw
黃宥綾	christina.ii12@nycu.edu.tw
林垣志	louislin.cs12@nycu.edu.tw
卓正修	cho312581013.ii12@nycu.edu.tw
施育衡	ra890927.cs12@nycu.edu.tw
羅翊軒	essen900718.cs12@nycu.edu.tw

Exercise Demo Rules

- Every week we have some simple exercises during the lab.
 - <https://nycu.webex.com/meet/TA-course>
- We will upload the slides on the Friday of the previous week.
- After the TA explanation, you can start to do your exercises.
- If you finish your exercises, raise your hand and wait for the TA to demo your exercises.
- Upload your exercises to E3.
- Format :
 - 311581003_ex_w01.zip (w01 means week 1)
 - 311581003_ex_01.cpp (01 means problem 1)
 - 311581003_ex_02.cpp (02 means problem 2)

Exercise Demo Rules

- If you are unable to complete the exercise on time or have a personal reason for not being able to attend class, you can demo next week with no penalty.
- Before midterm, you should complete all the exercises(including demo).
- If you can not attend the TA course, you should let us know.
- We may change the rules after midterm.

Homework Submission Rules

- After the Lab, there are some homework for you. Download them on E3.
- Format :
 - 311581003_hw_w01.zip (w01 means week 1)
 - 311581003_hw_01.cpp (01 means problem 1)
 - 311581003_hw_02.cpp (02 means problem 2)
- Upload your homework to E3, if there is format error, your score will get penalty **original score – 5**.
- Homework deadline : **before next TA class (Next Monday 23:55)**

Homework Submission Rules

- Late submission
 - $\text{score} \times 0.8$ within the first week(Monday 23:55)
 - $\text{score} \times 0.6$ within the second week(Monday 23:55)
 - score will be 0 after the second week
- We will execute your code on Visual Studio. If your code cannot run on it, the score will have discount with **original score – 30**.
- In summary, you can use any IDE which you like to write the code, but **make sure your code can run on Visual Studio!!!**
- **Plagiarism will result in a 0 score.**
- We may change the rules after midterm.

Introduce the C language and Print function

TA 顏廷恩

Download Visual Studio Community 2019

[download link](#)

Your Downloads and Product Keys

優點 下載 產品金鑰 訂閱 取得協助 市集

Product Family

- ☐ Agents for Visual Studio 2019
- ☐ Build Tools for Visual Studio 2019
- ☐ IntelliTrace Standalone Collector for Visual Studio 2019
- ☐ Performance Tools for Visual Studio 2019
- ☐ Remote Tools for Visual Studio 2019
- ☐ Visual C++ Redistributable for Visual Studio 2019
- ☐ Visual Studio 2019 for Mac
- ☐ Visual Studio Community 2019
- ☐ Visual Studio Enterprise 2019
- ☐ Visual Studio Professional 2019
- ☐ Visual Studio Team Explorer 2019

下載

- ☐ All Downloads
- ☒ Available for my subscription(s)

visual studio 2019

Product	Version		Download	Keys	Language	Architecture	Type	Release Date	Info
Visual Studio Enterprise 2019	16.11	▼	Download ↓	🔑	Multiple Languages ▼	x86 ▼	exe ▼	2023/8/8	📘
Visual Studio Team Explorer 2019	16.11	▼	Download ↓	🔑	Multiple Languages ▼	x86 ▼	exe ▼	2023/8/8	📘
Visual Studio Professional 2019	16.11	▼	Download ↓	🔑	Multiple Languages ▼	x86 ▼	exe ▼	2023/8/8	📘
Visual Studio Community 2019	16.11	▼	Download ↓	🔑	Multiple Languages ▼	x86 ▼	exe ▼	2023/8/8	📘
Agents for Visual Studio 2019	16.11	▼	Download ↓	🔑	English ▼	x64 ▼	exe ▼	2023/8/8	📘
Build Tools for Visual Studio 2019	16.11	▼	Download ↓	🔑	Multiple Languages ▼	x86 ▼	exe ▼	2023/8/8	📘
Visual C++ Redistributable for Visual Studio 2019	16.11	▼	Download ↓	🔑	Multiple Languages ▼	x64 ▼	exe ▼	2023/8/8	📘
IntelliTrace Standalone Collector for Visual Studio 2019	16.11	▼	Download ↓	🔑	English ▼	x86 ▼	exe ▼	2023/8/8	📘
Remote Tools for Visual Studio 2019	16.11	▼	Download ↓	🔑	English ▼	x64 ▼	exe ▼	2023/8/8	📘

Feedback

© 2023 Microsoft

需要協助您選擇安裝項目嗎? [更多資訊](#)

傳統型與行動裝置 (5)



.NET 桌面開發

使用 C#, Visual Basic 及 F#, 利用 .NET 和 .NET Framework 建置 WPF、Windows Forms 與主控台應用程式。



通用 Windows 平台開發

使用 C#, VB 或選用 C++, 來建立適用於通用 Windows 平台的應用程式。



使用 C++ 進行行動開發

使用 C++ 建置適用於 iOS、Android 或 Windows 的跨平台應用程式。



使用 C++ 的桌面開發

使用您選擇的工具 (包括 MSVC、Clang、CMake 或 MSBuild), 建置適用於 Windows 的新式 C++ 應用程式。



使用 .NET 進行行動開發

使用 Xamarin 建置適用於 iOS、Android 或 Windows 的跨平台應用程式。



位置

C:\Program Files (x86)\Microsoft Visual Studio\2019\Community [變更...](#)

繼續進行即表示您同意所選 Visual Studio 版本的[授權](#)。我們也可讓您使用 Visual Studio 下載其他軟體。此軟體為分開授權, 如同[協力廠商聲明](#)或其隨附的授權中所述。繼續進行即表示您也同意該授權。

安裝詳細資料

Visual Studio 核心編輯器

使用 C++ 的桌面開發

使用 C++ 進行行動開發

已包含

- ✓ C++ 核心功能
- ✓ Android SDK 安裝程式 (API 層級 25) (可用...

選擇性

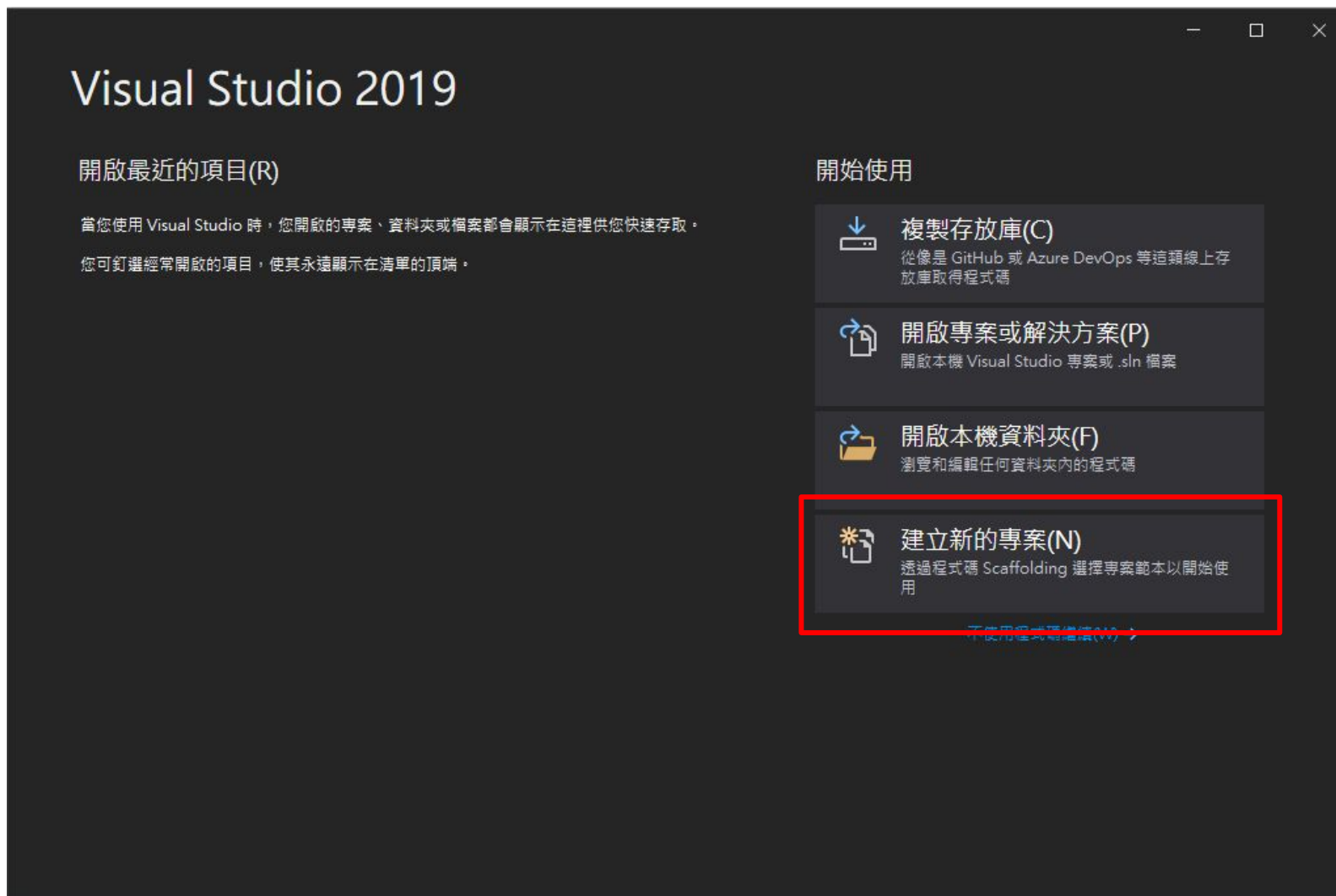
- ☒ Android NDK (R16B)
- ☒ Apache Ant (1.9.3)
- ☒ C++ Android 開發工具
- ☒ IntelliCode
- ☐ Google Android Emulator (API 層級 25) (本...
- ☐ Intel Hardware Accelerated Execution Man...
- ☐ Android NDK (R16B) (32 位元)
- ☐ C++ iOS 開發工具
- ☐ Incredibuild - 建置加速

總共所需空間 16.33 GB

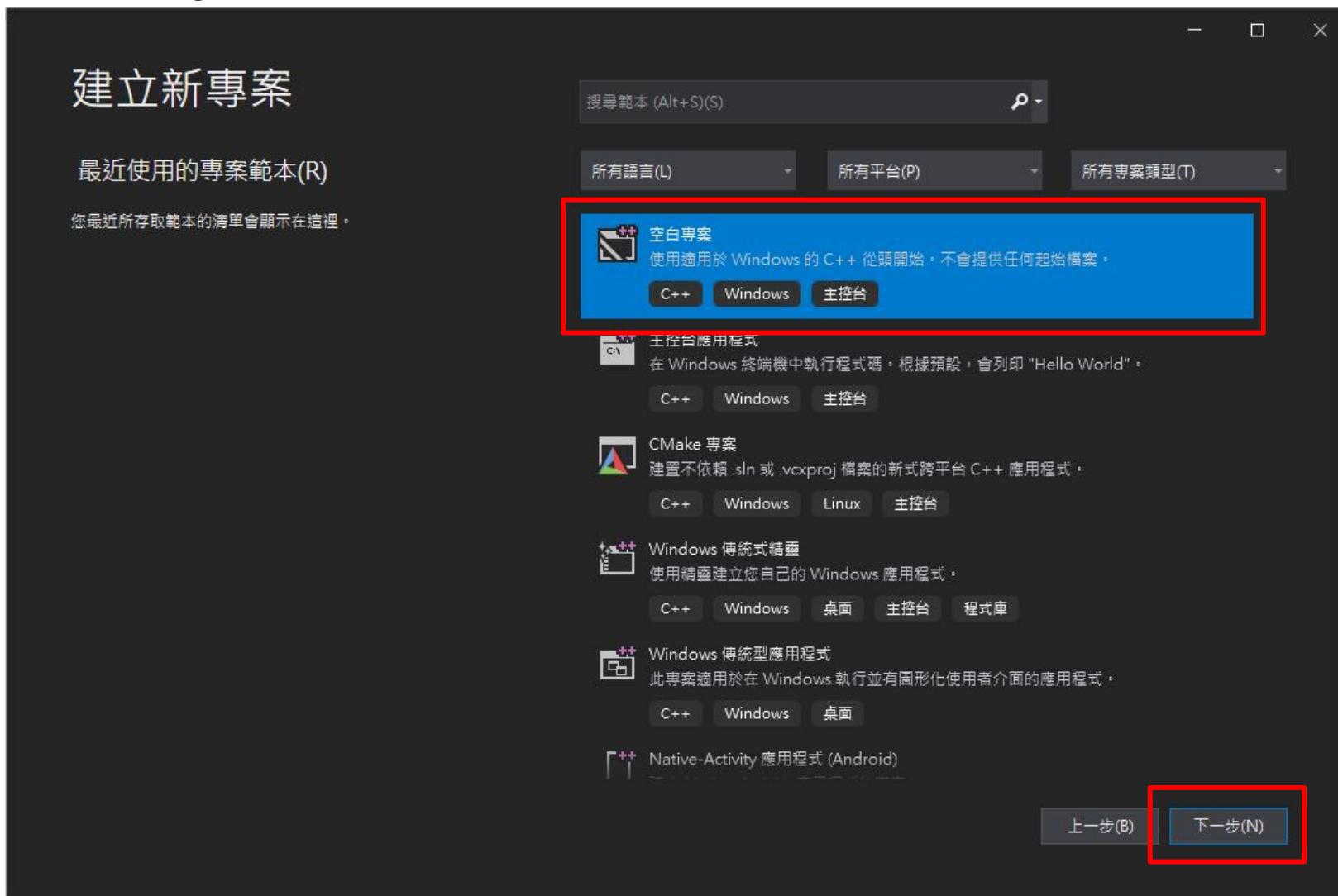
在下載時安裝

安裝(I)

Create Project



Create Project



Create Project

設定新的專案

空白專案 C++ Windows 主控台

專案名稱(J)

311581003_ex_w01

位置(L)

C:\Users\user\Desktop\C\ ...

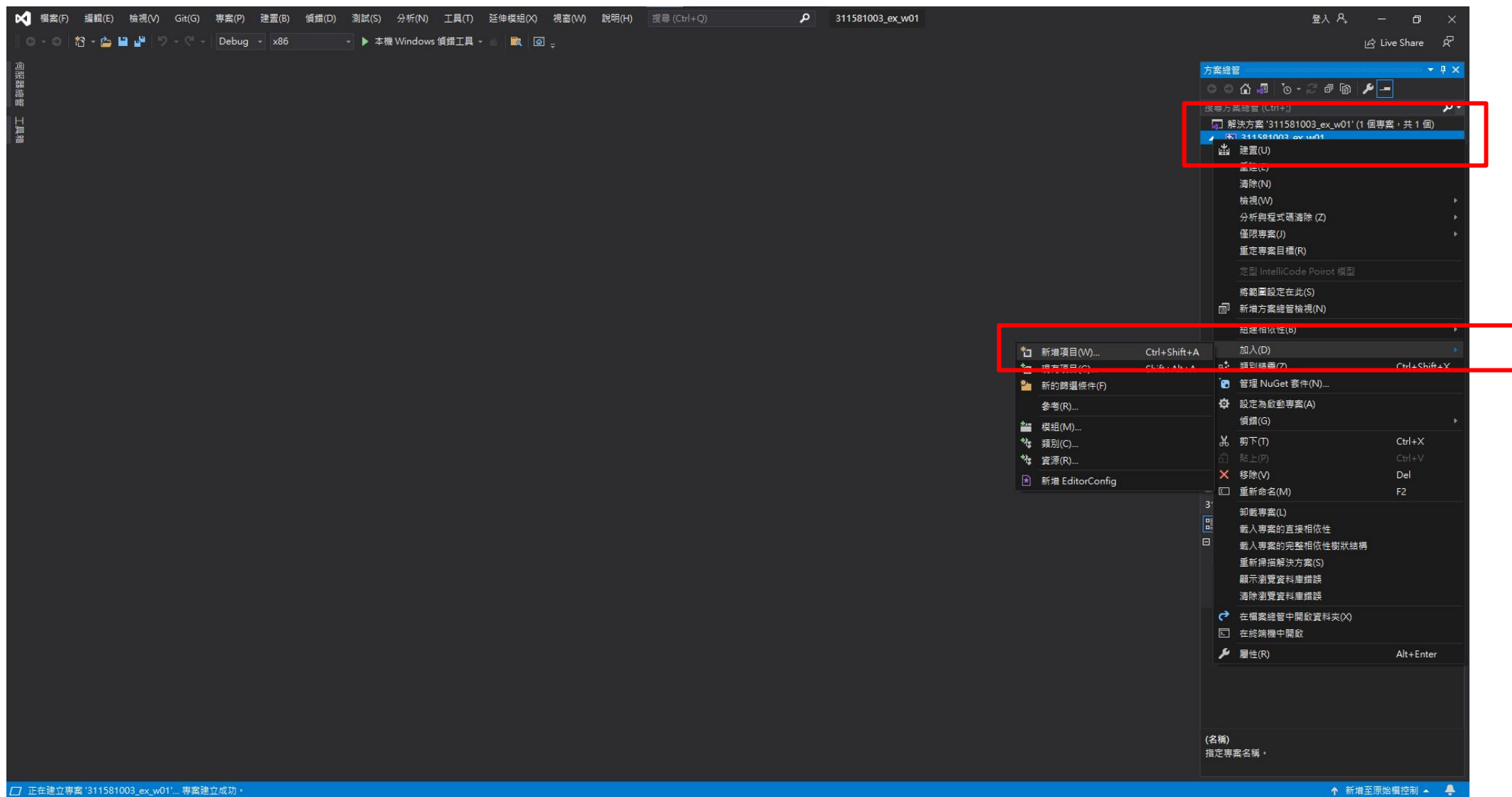
解決方案名稱(M) ⓘ

311581003_ex_w01

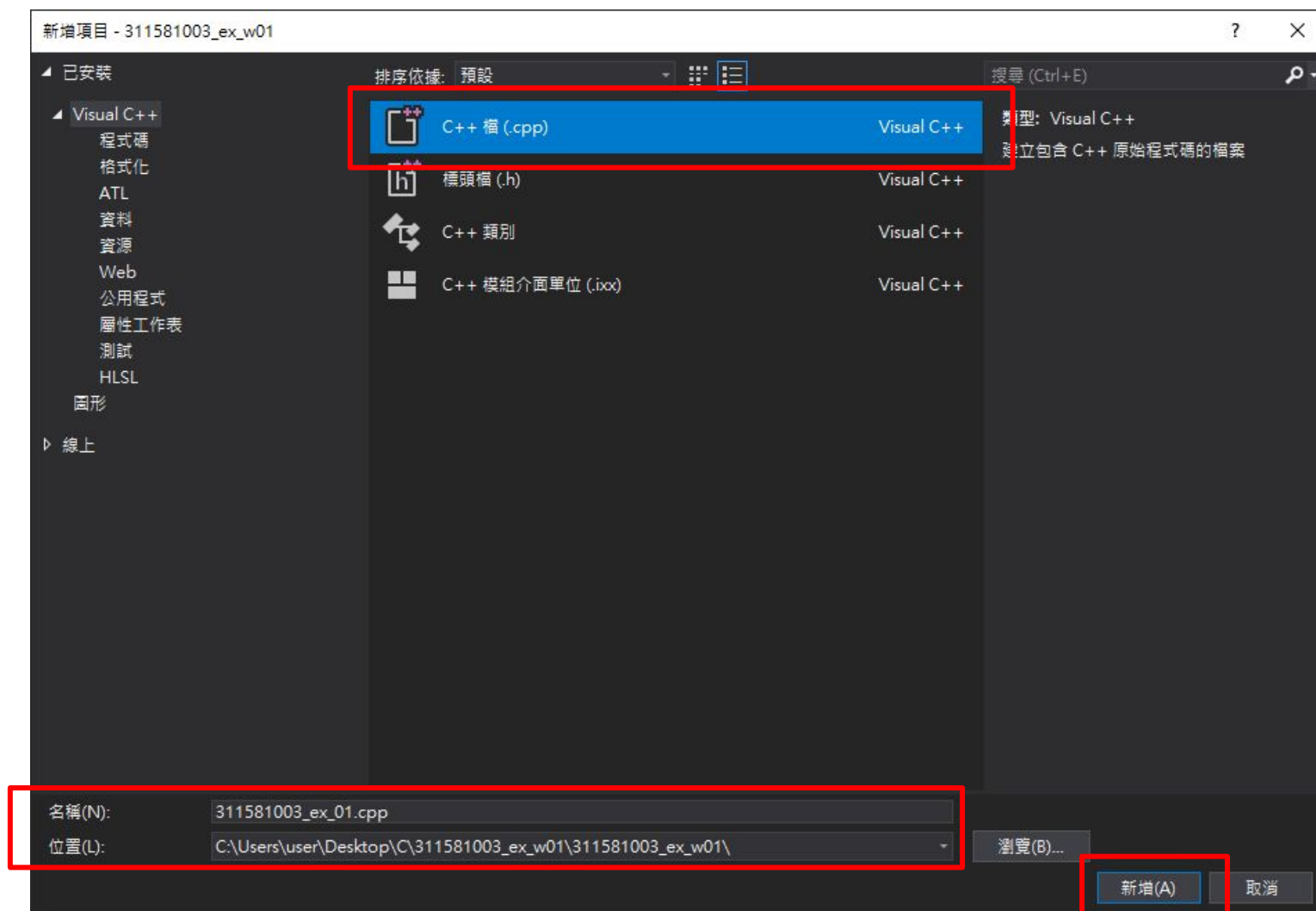
☐ 將解決方案與專案置於相同目錄中(D)

上一步(B) 建立(C)

Create project



Create Project

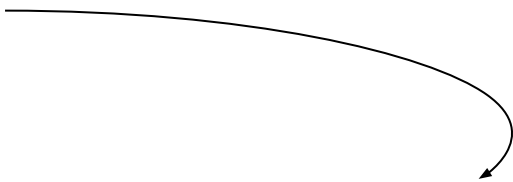


Hello World!

```
#include <stdio.h> // standard input/output
#include <stdlib.h> // standard library

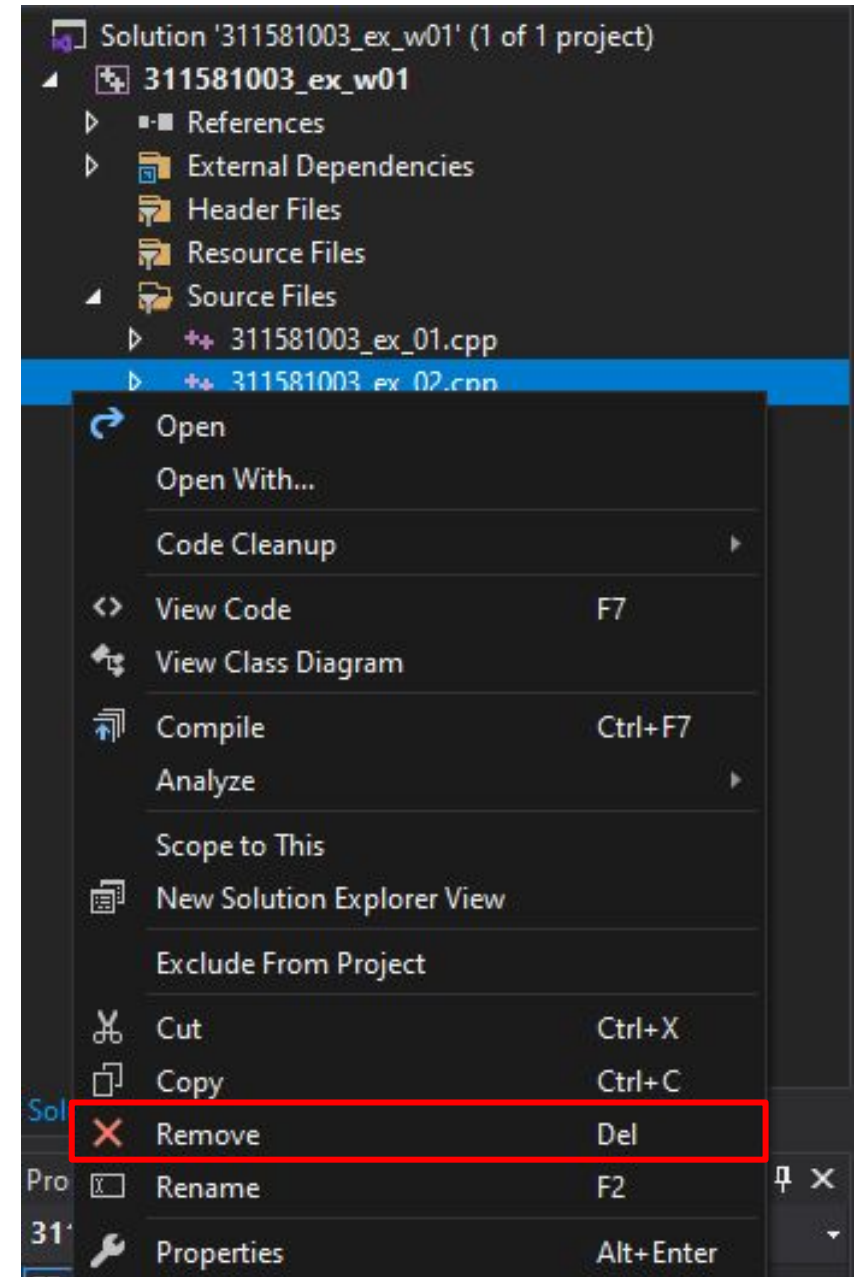
int main()
{
    printf("Hello World!\n");

    system("pause"); //pause the program
    return(0);
}
```



C:\Users\user\Desktop\C\311581003_ex_w01\Debug\311581003_ex_w01.exe
Hello World!
請按任意鍵繼續 . . .

For a single project, you can only execute one file at a time. Please remove the first file when working on the second exercise.



Basic

Standard Input/Output
Ex: scanf(), printf()

Standard Library
Ex: system()

Comment :

Single line comment : //

Multiple lines comment : /* */

```
#include <stdio.h> // standard input/output
#include <stdlib.h> // standard library

int main()
{
    printf("Hello World!\n");

    system("pause"); //pause the program
    return(0);
}
```

Main program :
Write your code in it

Basic

- The letters of the alphabet are regarded as different in case.
 - `Int` \neq `int` \neq `INT`
 - `Main` \neq `main` \neq `MAIN`
- Brackets must appear in pairs.
 - `()`、`[]`、`{}`
- Don't forget to add a semi-colon at the end of the sentence.
 - `printf("Hello World!");`

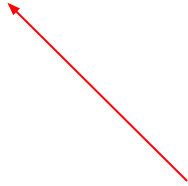
Variable

Data Type

Variable value

`int a = 10;`

Variable name



Common Data Type

Type	Sizes	Range
int	4 bytes	-2147483648 ~ 2147483647
float	4 bytes	$2.939 \times 10^{-38} \sim 3.403 \times 10^{+38}$
double	8 bytes	$5.563 \times 10^{-309} \sim 1.798 \times 10^{+308}$
char	1 bytes	-127 ~ 128

- For more data type, you can refer to the [link](#).

Variable name

- Composed of letters, digits, and the underscore character
 - Ex: `int *&^%` is not allowed
- Can not use reserved words
 - Ex: `int int` is not allowed
- Meaningful is better
 - Ex: `int sum_of_digits` is easy to verify the meaning of variable
- Can not begin with Chinese
 - Ex: `int 年齡` is not allowed
- Can not begin with a digit
 - Ex: `int 1NumberOfUserA` is not allowed

Variable value

- int

- int a = 10;

Note that the meaning of the variable is **a is assigned by the value 10**

- float

- float b = 1.1;

- double

- double c = 2.2;

- char

- char d = 'D' ;

printf()

- Print formatted data
- How to use?
 - Code : `printf("Hello World!");`
 - Output : Hello World!
 - Code : `printf("Hello\nWorld!");`
 - Output : Hello
World!
- How to print Hello\nWorld!?
- Answer : `printf ("Hello\\nWorld!");`

Escape sequence	Meaning
<code>\\</code>	<code>\</code> character
<code>\'</code>	' character
<code>\"</code>	" character
<code>\?</code>	? character
<code>\a</code>	Alert or bell
<code>\b</code>	Backspace
<code>\f</code>	Form feed
<code>\n</code>	Newline
<code>\r</code>	Carriage return
<code>\t</code>	Horizontal tab

printf()

- How to print an integer?
- Answer : `printf("%d", n);`

Print in decimal

Variable

Format	Meaning
%d	decimal
%o	octal
%u	unsigned integer
%x, %X	hexadecimal

- How to print a char?
- Answer : `printf("%c", c);`

A character variable

- Can following code run?

```
int a = 65;  
printf("%c", a);
```

- Answer is YES! It will output **A**
- You can refer to ASCII.

printf()

- How to print the floating point?
- Answer : `printf("%f", f);`

```
float f = 6.5;
printf("%f\n", f);
```

C:\Users\user\source\repos\Project1\De
6.500000
請按任意鍵繼續 . . .

- What if we want to print the floating point to the second decimal place?
- Answer : `printf("%.2f", f);`

```
float f = 6.5;
printf("%.2f\n", f);
```

C:\Users\user\source\repos\Project1\Debug
6.50
請按任意鍵繼續 . . .

printf()

- In short summary, the format tags prototype is :

`%[flags][width][.precision][length]specifier`

- Ex: float g = 3852.991;

```
printf(“*%+010.2f*\n”, g);
```

- Output : `*+003852.99*`

- There are more useful format for printf()! Refer to this [link](#)!

printf()

- We can print a lot of variables in a printf().
- Ex :

```
int a = 1;
float b = 2.0;
char c = 'C';
printf("%d, %.1f, %c\n", a, b, c);
```
- Output : 1, 2.0, C

scanf()

- Read the inputs.
- In Visual Studio, we must add **#define _CRT_SECURE_NO_WARNINGS** (add this at the first line)

scanf()

- How to use?
- Answer : `scanf("%d", &num);`



Point to the memory address where variable num is stored

scanf()

- We can get lot of variables in a scanf().

- Ex : int a;

float b;

```
scanf("%d%f", &a, &b);
```

```
printf("%d, %.1f\n", a, b);
```

- Input : 1 2.51233

Output : 1, 2.5

Arithmetic Operator

- Add, Subtract and Multiplication

- Ex : int a, b;

```
a = 10;
```

```
b = 20;
```

```
int c = a + b;
```

```
int d = a - b;
```

```
int e = a * b;
```

```
printf("%d, %d, %d", c, d, e);
```

- Output : 30, -10, 200

- Division and Remainder

- Ex : int f = 50;

```
int g = 20;
```

```
int h = f / g;
```

```
float i = (float)f / g;
```

```
float j = f / g;
```

```
int k = f % g;
```

```
printf("%d, %f, %f, %d", h, i, j, k);
```

- Output : 2, 2.500000, 2.000000, 10

Turn variable **f** from **int** to **float**

Arithmetic Operator

- +=、-=、*=、/=、%=

- Ex : int score = 20;

score += 10; // equal to **score = score + 10**

score -= 10; // equal to **score = score - 10**

score *= 10; // equal to **score = score * 10**

score /= 10; // equal to **score = score / 10**

score %= 10; // equal to **score = score % 10**

printf("%d", score);

- Output : 0

Arithmetic Operator

- `++` and `--` are equal to `+=1` and `-=1`, that is, `a++` is equal to `a += 1`.

- But there are some tricky parts, Ex :

- `int a = 1;`

```
printf("%d, ", ++a);
```

```
printf("%d, ", a++);
```

```
printf("%d, ", a--);
```

```
printf("%d", --a);
```

- Output : 2, 2, 3, 1

If `++` is at the right of the variable, it will change the value at next call.
In contrast, `++` at the left changes the value of the variable immediately.

Exercises

2023.09.12

Exercise 1

- Print a short self-introduction by following format.
 - Use a variable called age

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int age = 18;
7
8      C:\Users\Ting-En\Desktop\311581003_ex_w01\Debug\311581003_ex_w01.exe
9      Hello, I am Yen Ting-En.      I'm 18 years old.
10     I "have" learned programming languages.
11     Press any key to continue . . .
```

horizontal tab

new line

add " " in the word have or haven't

Exercise 2

- Input two integers, output add, subtract, multiple, divide and get the remainder of two integers. Note that you must make floating point to the **second decimal place**.

- Sample Input :
30 7

- Sample Output :
37
23
210
4.29
2