Introduction to C++

Tai, Wei Hsuan

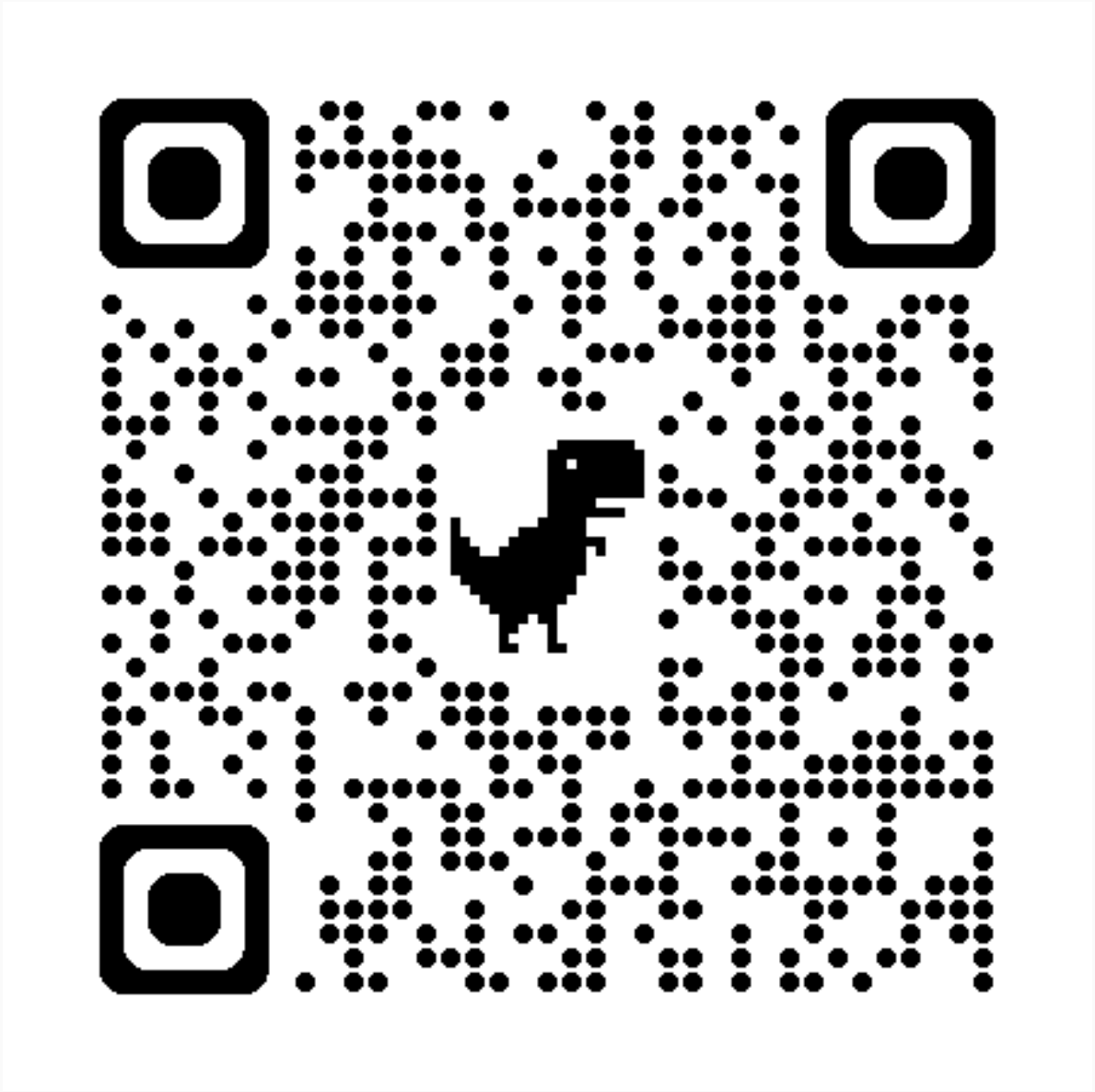
week 1

Outline

1. 課程介紹
2. Why C++?
3. What is C++?
4. Basic Structure of C++ Program
5. Standard Output in C++
6. Standard Input & Variables in C++

課程介紹

課程簡報



關於我...

- 戴偉璿（Tai, Wei Hsuan）
- 台大醫工大三
- 擅長：C++、資料結構與演算法、網際網路概論、網站前後端架設、Linux、機器學習...
- 興趣：寫程式、看棒球、打電動
- 張學友的粉絲

進度安排

日期	主題	日期	主題
9/12	課程簡介、基礎輸入輸出、變數	11/28	函式
9/19	變數的運算	12/5	遞迴
10/17	選擇結構與邏輯運算子	12/12	Struct
10/31	重複結構	12/19	Vector
11/7	字串處理	12/26	Stack, Queue
11/21	期中考	1/2	Set, Map, Priority Queue
		1/9	期末考

上課方式

- 觀念講解、範例示範
- 大量的練習題
- 大量的數學證明（如果有必要）
- 期中考與期末考
- Zerojudge Code: `DXg60o`

關於 LLM

- Large Language Model
- ChatGPT、Grok、Claude、Gemini
- 以討論取代抄答案
- **請不要叫我檢查你用 LLM 寫的程式碼！**

Why C++?

當亞洲父母發現你學的

是C++而不是A++

C++ vs Python

- C++:
  - Compiled Language
  - Static Typing
  - High Performance
  - Working Close to Hardware
- Python:
  - Interpreted Language
  - Dynamic Typing
  - Easy to Learn and Use
  - Rich Libraries and Frameworks

C++ is everywhere!

- Operating Systems（Windows、macOS、Linux）
- Web Browsers（Chrome、Firefox、Edge）
- Game Engines（Unreal Engine、Unity）
- Parallel Computing（CUDA、OpenCL）

What is C++?

Introduction to C++

- First released in Nokia Bell Labs by Bjarne Stroustrup
- C with Classes
- 1983s, C with Classes → C++
- Object-Oriented Programming
- Directly manipulate hardware resources(memory, CPU)

”Father” of Programming Languages

- CPython, Numpy, Pandas, Matplotlib, Scikit-learn, TensorFlow, PyTorch
- JavaScript V8 Engine, Node.js, Deno
- MySQL, PostgreSQL, MongoDB
- Even C++ compiler itself!

Simple C++ IDE

- Windows: Dev-C++
- macOS: Xcode

Basic Structure of C++ Program

Hello World!

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      cout << "Hello World!" << '\n';
6      return 0;
7  }
```

Three Main Parts

- Header Files
- Namespaces
- Main Function

Header Files

In python, we use `import` to include libraries.
In C++, we use `#include` to include header files.
The most common header file is `<iostream>`, which is used for input and output operations(io refers to input/output). But if you want to use other advanced features, you may need to include other header files(i.e. `<vector>`, `<algorithm>`). Therefore, we often use `#include <bits/stdc++.h>` to include most standard libraries in contests(Do not do this in production code!).

Namespaces

Header file may contain many functions, classes, and variables. To avoid name conflicts, C++ uses namespaces to group related code together.

Here's an example: There's a function called `sort` in the standard library, and you may also define your own function called `sort`.

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  void sort(int arr[], int n) {/*sort function here*/}
5
6  int main() {
7      int a[] = {3, 1, 2};
8      sort(a, 3);
9  }
```

The code above may cause "ambiguous call" error.

Therefore, we need to use an another namespace to wrap our own code.

```
1        namespace my_sort {
2            void sort(int arr[], int n) {/*sort function here*/}
3        }
```

Or a much easier way, do not use the same name as standard library.

Main Function

In your code, you may define lots of functions, global variables, and classes. To tell the compiler where to start executing your program, you need to define a `main` function.

The `main` function is the entry point of your program. You can call other functions and use global variables inside the `main` function.

The concept of main function is similar to `if __name__ == "__main__":` in Python.

Look back to Hello World!

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

int main() {
    cout << "Hello World!" << '\n';
    return 0;
}
```

Practice 1: Zerojudge d483.

Standard Output in C++

cout

After the explanation of main function, we can now understand much part of it. However, what is `cout`. `cout` is the standard output stream in C++. It is used to print output to the console.
Its function is similar to `print()` in Python.
You can output string, integer, float, and other data types using `cout`.
If you want to print multiple items, you can use the insertion operator `<<` to chain them together(To memory the direction of `<<`, think about the flow of data).
If you want to output a pure variable or number, you can just use `cout<<variable;`, but if you want to output a string, you need to put it in double quotes(`" "`). More, if you want to output a single character, you need to put it in single quotes(`' '`).

Example of cout

Here's an example of using `cout`:

```
1          int main(){
2              string name = "World";
3              cout<<"Hello"<<name<<"!"<<'\n';
4          }
```

In this case, name is a variable that stores the string "World". What do you think the output will be?

Example of cout

It will be `HelloWorld!`. Because there's no space between `Hello` and `World!`.

If you want to add a space, you can do it like these:

```
1        cout<<"Hello "<<name<<"!"<<'\n';
2        cout<<"Hello"<<' '<<name<<"!"<<'\n';
```

Practice 2: Zerojudge a001.

Escape Characters

Sometimes if you want to print a new line, tab, or other special characters (i.e. \), you can use escape characters.

code	Character
`\n`	new line
`\t`	Tab
`\v`	vertical Tab
`\"`	”

TL;DR: If you can’t print a special character directly, just add a \ before it.

Practice 3: Zerojudge e926.

Standard Input & Variables in C++

Variables in C++

Variables are used to store data in your program.
In Python, you may define a variable like this:

```
1        name = "World"
2        age = 18
```

This is why python is called ”weakly typed language”.
In C++, you need to specify the data type of the variable when you define it.

```
1        string name = "World";
2        int age = 18;
```

Data Types in C++

Here are some common data types in C++:

Table 1: Common C++ Fundamental Types

Type	Typical Range	Size (bytes)
`int`	$-2^{31} \sim 2^{31} - 1$	4
`long long`	$-2^{63} \sim 2^{63} - 1$	8
`float`	$\pm 1.2 \times 10^{-38} \sim 3.4 \times 10^{38}$	4
`double`	$\pm 2.3 \times 10^{-308} \sim 1.7 \times 10^{308}$	8
`char`	$-2^{7} \sim 2^{7} - 1$	1
`bool`	0 or 1	1

Input to a Variable

You can use `cin` to get input from the user. For example:

```
1        int age;
2        cin >>age;
3        cout<<"You are "<<age<<" years old."<<'\n';
```

You can also get multiple inputs in one line:

```
1        int dd, mm, yyyy;
2        cin >>dd>>mm>>yyyy;
3        cout<<"Your birthday is "<<dd<<"/"<<mm<<"/"<<yyyy<<'\n';
```

ASCII

In the previous table, you may notice that `char` stores a number instead of a character.
This is because characters are represented by numbers in the computer. The most common character encoding is ASCII.
For example, the character 'A' is represented by the number 65 in ASCII.
You can use `int('A')` to get the ASCII value of 'A' in C++.

0	NUL	16	DLE	32		48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(	56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41	)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[	107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	|
13	CR	29	GS	45	-	61	=	77	M	93	]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL