# Introduction to C++

Tai, Wei Hsuan

week 1

# 課程介紹

# 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 | 11/28 | 函式 |
| 9/19 | 變數與四則運算 | 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）

# 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++ 就藏在生活中！

- 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!" << endl;
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!

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

# Standard Input and Output in C++

# cout

After the explanation of main function, we can now understand much part of it. However, what is `cout` and `endl`?

`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).

# Example of cout

Here's an example of using `cout`:

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

# Escape Characters

In the previous codes, you may find `endl` appears frequently.

In fact, `endl` is an escape character that represents a new line.

Other common escape characters include: