

# C++ 簡介

## 在開始之前

- 推薦的參考工具書
  - C++ Primer,  
by Stanley B. Lippman.
- C++ 是一個宏偉複雜的語言，漸趨成熟、  
但仍然在發展中。（C++11, C++14, C++17）

## 在開始之前

- C++ 支援 C 語言絕大部份的 feature
  - 大部份情況下，  
可以當成進階版的 C 使用
- 在本課程裡介紹的內容，以基礎概念以及函式庫的工具實際使用為主，不牽涉到太深的語法。

## 基礎的 C++ 程式框架

```
#include <bits/stdc++.h>

// 引用所有 C++ 的標準 header file

using namespace std;

// 引用所有定義在 std 命名空間底下的
// 識別字 (將它們納入當前的scope)
```

## 基礎資料型態

- 除了 C 已有提供的基礎資料型態之外
  - 整數、浮點數、字元
- C++ 提供了 Boolean 基礎資料型態
  - 以關鍵字 `bool` 代表,  
其值可為 `true` 或 `false`

## 抽象資料型態

- 除了四種類型的基礎資料型態之外，  
C++ 提供了一個豐富的程式庫，內含許多抽象資料型態 (Abstract Data Types)
  - 能用抽象方式描述、不牽涉到底層實際型態的「儲存資料的方式」(或是「資料結構」)
  - 例如：「陣列」是一種抽象的型態
  - 在 C++ 的程式庫裡，  
這些抽象資料型態以物件的方式呈現、實作

## 物件 (Object)

- 主要組成為兩個部份：
  - 屬性 (Attribute) :  
用來描述此物件的特徵、屬性、性質等。
  - 方法 (Method) :  
針對此物件的操作 (函式)、  
專屬於此物件的演算方法等。
- 可以討論物件的從屬、繼承關係等。

## C++支援的抽象陣列型態 – Vector

- Vector 是 C++ STL 函式庫提供的抽象資料型態，對應到「陣列」這種線性的儲存方式。
  - 特點是，當我們使用 `vector` 時，不需要考慮儲存空間大小的問題（它的大小可以隨著需要，自動增長）
  - 除此之外，`vector` 也支援傳統的陣列索引運算子



## C++支援的抽象陣列型態 – Vector

- Vector 提供的操作 (成員函式)
  - `size()` – 傳回 vector 儲存的資料數
  - `push_back(data)` –  
將 data 放到 vector 最後面的位置
  - `back()` – 傳回最後面的資料(reference)
  - `pop_back()` – 移除最後面的資料