1

CT3407302

C/VB程式與應用

C/VB Programming and Applications

2

為什麼要學習程式寫作?

Why Programming ?

3

為什麼要學習程式寫作？

- 程式寫作能力 vs. 解決問題之能力
- 可以隨心所欲地應用電腦解決問題
  - 電腦硬體 ← 電腦軟體 ← 程式
- 其它
  - 建立自己的第二專長、了解電腦應用程式背後的邏輯、充份利用電腦之功能 (運算、儲存、傳輸) ...

4

- It has often been said that a person does not really understand something until he teaches it to someone else.
- Actually a person does not really understand something until after teaching it to a computer.

~ Donald Knuth

5

為什麼學 C/C++

Why C/C++

6

為什麼學 C/C++

- 所有現代的程式語言 (C#, Java, Javascript, etc.) 的語法都是基於 C/C++ 。
  - 如果你有興趣寫手機程式或遊戲
    - iPhone / iPad → Objective C (一種 C 語言)
    - Android → Java
- 所有重要的程式設計觀念都可以介紹到

7

課程目的

學習電腦程式之寫作以
- 強化解析問題與電腦應用之能力;
- 進一步了解電腦程式運作之原理;
- 期望將來能充份運用電腦運算、資料儲存與傳輸之能力協助解決工程問題。

8

課程大綱

- 基本程式架構
- 變數與算術運算
- 程式流程控制
- 迴圈與陣列
- 函式與標準函式庫
- 多維陣列、指標、與動態記憶體配置
- 變數之儲存等級與函式呼叫之參數傳遞
- 函式覆載與遞迴函式

9

課本

無指定課本，以兩班的上課講義為主。
http://140.118.105.174/Courses/CVB/ (also on BB)
http://hungming.ct.ntust.edu.tw/2010spring/ct3407/ct3407.htm

但請各位同學自己選定一本自己最能接受的 C++ 書籍。

儘量不要找 Visual C++、視窗程式設計的書，因為其內容大部份都不會用到 ...

10

使用之開發環境

Microsoft Visual Studio 2008 Express

http://www.microsoft.com/express/Downloads/#2008-Visual-CPP

http://140.118.105.195/Courses/CVB/VS2008.iso

11

學期成績

30%: 期中考

30%: 期末考

30%: 作業

10%: 課堂參與

12

作業成績

作業遲交一星期以九折計算。

遲交超過一星期以零分計算。

抄襲以零分計算?!

13

答疑時段?

Lecture 1

基本程式架構
變數與算術運算

15

今日大綱

- Visual Studio 2008 Express 使用
- 第一個程式
- 程式如何被執行
- 變數 (variables)

16

Microsoft Visual Studio 2008 Express 使用

17

操作 (I)

- File → New Project → Win32 Console Application → 輸入一個專案名稱→ OK
- Next → Console Application/Empty Project → Finish
- 在原始程式檔案 “Source Files” 上按右鍵 → Add → New Item → C++ File (.cpp) → *ex01.cpp* → Add
- 現在可以開始寫程式囉，完成後按 CTRL-F5 (或 Debug → Start without Debugging) 即可開始執行。

18

第一個程式

The first C++ program

19

1-1.cpp

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

int main() {
    // 輸出 "Hello World" 至螢幕上
    cout << "Hello World";

    /* 函式執行完畢，回傳 0 */
    return 0;
}
```

此被稱為是原始程式碼 (source code)

20

觀察

- 程式由許多的敘述 (statement) 組成
- C/C++ 的敘述以分號 (semi-colon) 作為結尾
- 有兩行我們看得懂的部份為給人看的註解 (comment, remark)
- 有幾行看似英文的敘述即為程式碼的部份

21

#include <iostream>

- 目的：將 iostream 這個標頭檔 的內容「含括」(include) 進來。
  - 標頭檔 (header files)：定義了一些可使用的函式 (function) 或物件 (object)
  - iostream: 定義了以「串流」(stream) 進行輸入與輸出的物件
    - cout: **c**onsole **out**put - 從螢幕輸出字元/字串
    - cin: **c**onsole **in**put - 從鍵盤輸入字元/字串

22

using namespace std;

- **目的**：告訴電腦在尋找一函式或物件時，可在 std (standard, 標準) 命名空間內搜尋。
- **說明**：現在對此不清楚沒有關係，我們本學期的程式大多都會需要此行敘述。

23

主程式區塊
main program block

```
int main() {

  … 主程式內容 …

  return 0;
}
```

{ ….} 定義了一個區塊

`int main() {…}` 定義了主程式區塊，此部份以後會再更詳細的介紹，目前僅需要記得整個 C/C++ 的主程式定義如上所示即可。

24

C/C++ 程式註解

單行註解
範例：`// output "Hello World" to console`
說明：`//` 之後所有的字都被認為是註解的一部份直到換行為止

區塊註解
範例：`/* return value 0 */`
說明：
- /* 表示註解區塊的開始
- */ 表示註解區塊的結束
- 可用來註解多行

```
/**********
  這是一個註解
  這裡還是註解
  這行還是註解
  **********/
```

```
/* 這也是一行註解 */
```

```
// 這也是一行註解
```

```
// 這也是一行註解 //
```

25

為什麼要寫註解？
Why write comments or remarks?

- 讓 programmer 提醒自己程式某片段程式的功能
- 讓別人比較有機會看得懂自己寫的程式
- 讓程式變得比較好維護
- 讓自己以後有機會看懂自己的程式

26

cout << "Hello World";

- 目的：將 “Hello World” 這個「字串」(string) 輸出至螢幕上。
- 說明：
  - cout: console output, 為一個「物件」 (object)，用來將接收到的資料輸出至螢幕上。
  - <<: 串流運算子 (stream operator)，用來將右邊 (RHS, Right Hand Side) 的資料傳送到左邊 (LHS, Left Hand Side) 的物件。
  - “Hello World”: 為字串資料，在C/C++中，**字串**必需使用**雙引號**括起來。

27

1-1.cpp

```
#include <iostream>
using namespace std;
//
int main() {
    // 輸出 "Hello World" 至螢幕上。
    cout << "Hello World";

    /* 函式執行完畢，回傳0。 */
    return 0;
}
```

28

變數
Variable

電腦具有處理大量資料的能力, 但是資料必需先存在記憶體中, 而變數就是用來幫助我們存取記憶體的工具...

29

變數
Variable

- 記憶體是一連串 0 與 1的組合 → bit
- 電腦的記憶體單位
  - Bit → 0, 1
  - Byte → 8 bits → 2^8=256 → 0 – 255 or -128 – 127
- 在高階語言裡，我們使用變數來**利用記憶體資源**，用來進行資料的儲存與計算。
- 一個變數具有：
  - **名稱** (identifier): 用來區別不同的資料存放位置
  - **型態** (type): 代表該變數所儲存資料的意義

30

1-2.cpp

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

int main() {
  int a = 3;
  int b, c;

  b = 4;
  c = a + b;
  cout << "a+b=" << c;

  return 0;
}
```

宣告一個變數 a, 並給予初始值 3
宣告兩個變數 b 與 c

令 b 變數的值為 4
令 c 變數的值為 a+b 的計算結果
輸出 字串 “a+b=“ 以及變數 c 的值

a+b=7

31

說明

- 所有變數在使用**前**必需先被宣告 (declaration)
  - int a = 3; ← 宣告一個變數, 名稱為 a, 並賦予初始值為 3
  - int b, c; ← 宣告兩個變數, 名稱為 b 和 c。
  - 在 C 語言時代, 所有變數都必需在所有的敍述**之前**宣告完成。在 C++，變數宣告只要在使用前宣告即可，沒有一定要放在程式的開頭。
- b = 4;
  - 此行敍述 (statement) 執行了指定 (assignment) 的動作，將右側的運算元 (operand) 之值指定給左側的運算元。
  - 指定 (=) 為一種二元運算子 (binary operation) – 需要兩個運算元的動作或計算。
  - 亦即，將值 4 指定給變數 b、或令 b 的值為 4。

32

說明

- c = a + b;
  - 此行敍述中共有兩個二元運算子，第一個為 + 、第二個為 =
  - 由於運算子之優先順序 (precedence) 的緣故，+ (加法運算) 先執行、= (指定) 運算後執行。
  - 此行敍述即將 a + b 的值計算完後，將其和 (sum) 指定給 c 。
- `cout << "a+b=" << c;`
  - 此行敍述 (statement) 中， << 為資料流插入運算子，目的在螢幕上輸出 **字串** "a+b="與 變數 c 之值。

33

變數名稱
name, identifier

- 用來區別不同的變數，且大小寫視為不同 (case sensitive)
- 文字、數字、與底線符號的組合
- 第一個字元必需為文字或是底線
  - 正確: myName, _myName2, my_Name5,
  - 錯誤: 4Name, 4_name, my-Name
- 變數名稱不可為保留字 (reserved words), 如C/C++語言中的指令 while, for, if, ...
- 慣例:
  - 名稱盡量取得有意義: myName, my_name
  - *由底線開頭的名稱保留給各函式庫內部之變數*
  - *名稱全部大寫保留給常數變數*

34

正確或錯誤的變數名稱?

__name
8dogs
myGoodFriend
MyDog
CSI
www.ntust.edu.tw
ABcdEFG
Iam_28_yearsOld
13Ghosts

35

1-2.cpp

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

int main() {
  int a=3;
  int b, c;

  b = 4;
  c = a + b;
  cout << "a+b=" << c;

  return 0;
}
```

36

變數型態
variable type

意義	變數型態	記憶體量*	數值範圍	準確度 (Accuracy)
字元	char	1	-128 ~ +127	精確
整數	short *int*	2	-32768 ~ 32767	精確
	int	4	$-2^{31} \sim 2^{31}-1$	精確
	long *int*	4	$-2^{31} \sim 2^{31}-1$	精確
浮點數	float	4	$-3.4\times10^{38} \sim +3.4\times10^{38}$	6-7 位有效數字
	double	8	$-1.7\times10^{308} \sim +1.7\times10^{308}$	14 – 15 位有效數字
	long double	8	$-1.7\times10^{308} \sim +1.7\times10^{308}$	14 – 15 位有效數字

* 單位為 bytes。記憶體量與所使用之作業系統、硬體、以及編譯器等皆有關係，本表則依據win32作業系統之 Visual C++ 9.0 。

37

變數型態
variable type

- 變數型態決定了記憶體之使用量, 及該資料的意義
- 字元型態 (**char**acter)
  - char: 佔 1 byte = 8 bits的空間 → 2^8=256 種可能
  - char 實質上是整數, 電腦使用代碼來代表人們的文字
    - ASCII: American Standard Code for Information Interchange
  - Example: 65→'A', 66→'B', 97→'a', ...
  - http://www.asciitable.com/
  - unsigned char: 0 – 255
  - signed char: -128 – 127
- Example:

```
char myScore='A';
```

38

上傳!

- 請寫一個程式,
  - 宣告三個變數分別儲存你的出生年、月、日
  - 逐行印出自己的姓名、學號、班級，並利用之前宣告的變數輸出你的出生年月日，
  - 並印出其它你/妳想要透露給我或是助教的資訊 (e.g. 要 all pass ...)
- 完成後上傳至 BlackBoard