

C++程式設計基礎

lesson 1

陳毅

講師資訊

- 陳毅
- 台大資工所碩士
- 研究領域：Human computer interaction (人機互動)
- 擅長語言：C/C++, python, C#, Java
- 經歷：日本 SEGA 實習生
- 信箱：ck1001099@gmail.com



為什麼要學C++？

學習目標

- 能夠獨自撰寫簡易的C++程式
- 了解基礎的物件導向程式設計方法
- 閱讀程式碼的能力

課程規劃

- 8/4 – 8/20 每週一三五 18:10-21:00，共 8 堂
 - C++語法介紹
 - 範例練習與實作
- 評分標準
 - 100% 出席率：8次，一次12.5分。
- 課程資料：https://github.com/ck1001099-Teaching/cpp_course_2021summer

時間表

- | | | |
|--------|---|----------------------------|
| • 8/4 |] | • 資料型態、變數、基本輸入輸出 |
| • 8/6 | | • 流程控制（條件控制、迴圈控制） |
| • 8/9 |] | • 函式與陣列 |
| • 8/11 |] | • 指標 |
| • 8/13 | | |
| • 8/16 |] | • 資料結構
• 物件導向程式設計基礎 (2) |
| • 8/18 | | |
| • 8/20 | | |

學習資源

- ZeroJudge: <https://zerojudge.tw/>
- HackerRank: <https://www.hackerrank.com/>
- Leetcode: <https://leetcode.com/>
- CODESIGNAL: <https://app.codesignal.com/>
- FreeCodeCamp: <https://www.freecodecamp.org/>

概論

程式語言的演進

- 機器語言程式(Machine Code)
- 組合語言程式(Assembly Language)
- 高階語言程式：BASIC、COBOL、FORTRAN
- 結構化程式：Pascal、C
- 物件導向程式：C++, Java, C#

機器語言

```
1 FDX 12:01a 23- 1
A 002000 C2 30 REP #$30
A 002002 18 CLC
A 002003 F8 SED
A 002004 A9 34 12 LDA #$1234
A 002007 69 21 43 ADC #$4321
A 00200A 8F 03 7F 01 STA $017F03
A 00200E D8 CLD
A 00200F E2 30 SEP #$30
A 002011 00 BRK
A 2012

r
PB PC NUmxDIZC .A .X .Y SP DP DB
; 00 E012 00110000 0000 0000 0002 CFFF 0000 00
g 2000

BREAK

PB PC NUmxDIZC .A .X .Y SP DP DB
; 00 2013 00110000 5555 0000 0002 CFFF 0000 00
m 7f03 7f03
>007F03 55 55 00 00 00 00 00 00 00 00 00 00 00 00 00:UU.....
```

程式語言的演進

- 機器語言程式(Machine Code)
- 組合語言程式(Assembly Language)
- 高階語言程式：BASIC、COBOL、FORTRAN
- 結構化程式：Pascal、C
- 物件導向程式：C++, Java, C#

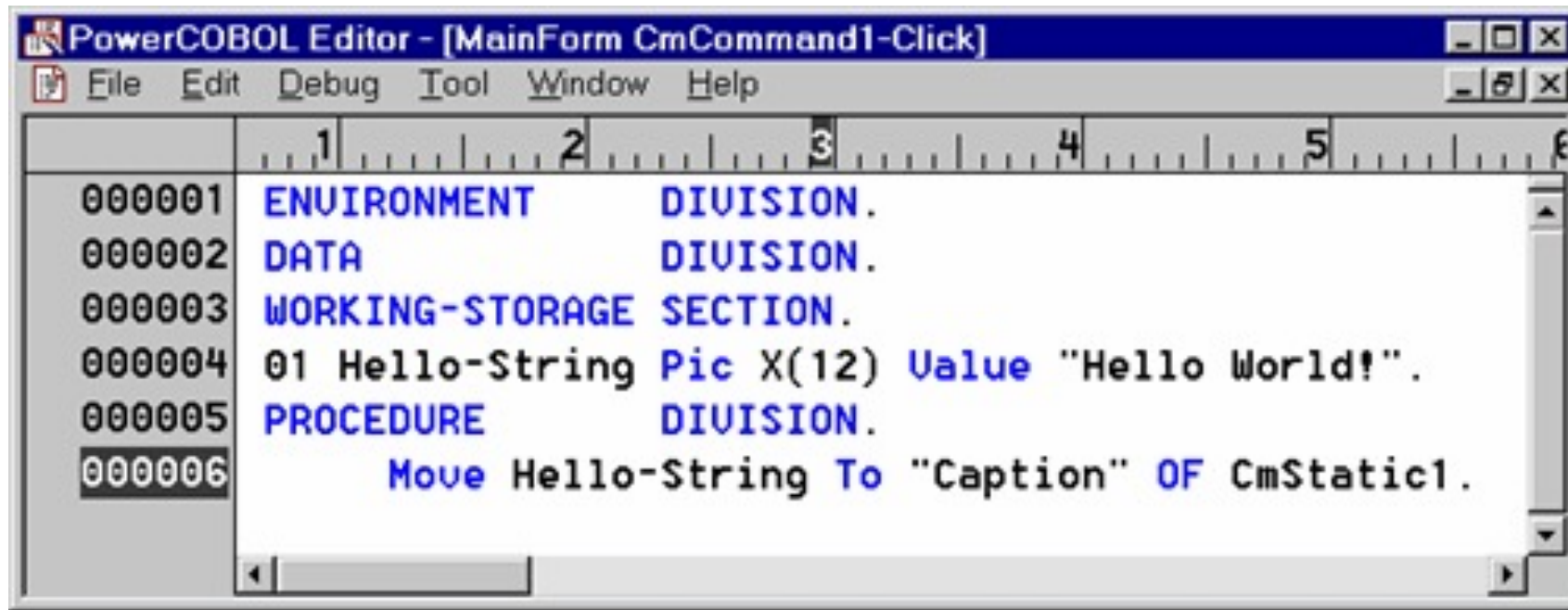
組合語言

```
1  section      .text
2  global      _start          ;must be declared for linker (ld)
3
4  _start:
5                                ;tell linker entry point
6      mov     edx,len          ;message length
7      mov     ecx,msg          ;message to write
8      mov     ebx,1            ;file descriptor (stdout)
9      mov     eax,4            ;system call number (sys_write)
10     int     0x80             ;call kernel
11
12     mov     eax,1            ;system call number (sys_exit)
13     int     0x80             ;call kernel
14
15  section      .data
16
17  msg         db  'Hello, world!',0xa ;our dear string
18  len         equ $ - msg          ;length of our dear string
```

程式語言的演進

- 機器語言程式(Machine Code)
- 組合語言程式(Assembly Language)
- 高階語言程式：BASIC、COBOL、FORTRAN
- 結構化程式：Pascal、C
- 物件導向程式：C++, Java, C#

高階語言 Cobol



The screenshot shows the PowerCOBOL Editor window with the title bar "PowerCOBOL Editor - [MainForm CmCommand1-Click]". The menu bar includes File, Edit, Debug, Tool, Window, and Help. The code is displayed in a text area with a line number column on the left and a ruler at the top. The code defines the ENVIRONMENT, DATA, and WORKING-STORAGE sections, and a PROCEDURE DIVISION containing a single statement to move the value of "Hello-String" to the "Caption" property of "CmStatic1".

```
000001 ENVIRONMENT      DIVISION.  
000002 DATA              DIVISION.  
000003 WORKING-STORAGE SECTION.  
000004 01 Hello-String Pic X(12) Value "Hello World!".  
000005 PROCEDURE          DIVISION.  
000006     Move Hello-String To "Caption" OF CmStatic1.
```

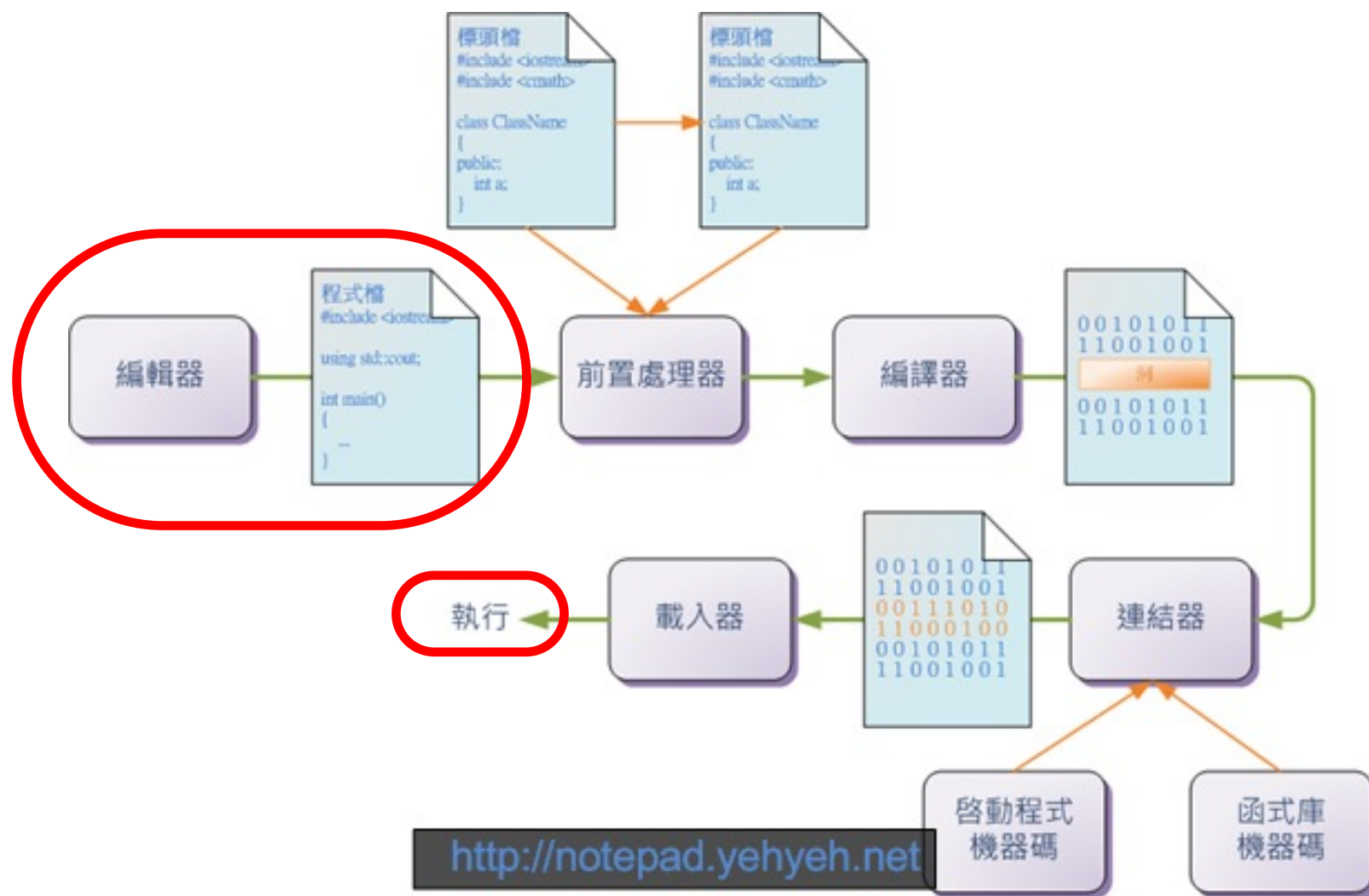
程式語言的演進

- 機器語言程式(Machine Code)
- 組合語言程式(Assembly Language)
- 高階語言程式：BASIC、COBOL、FORTRAN
- 結構化程式：Pascal、C
- 物件導向程式：C++, Java, C#

C++

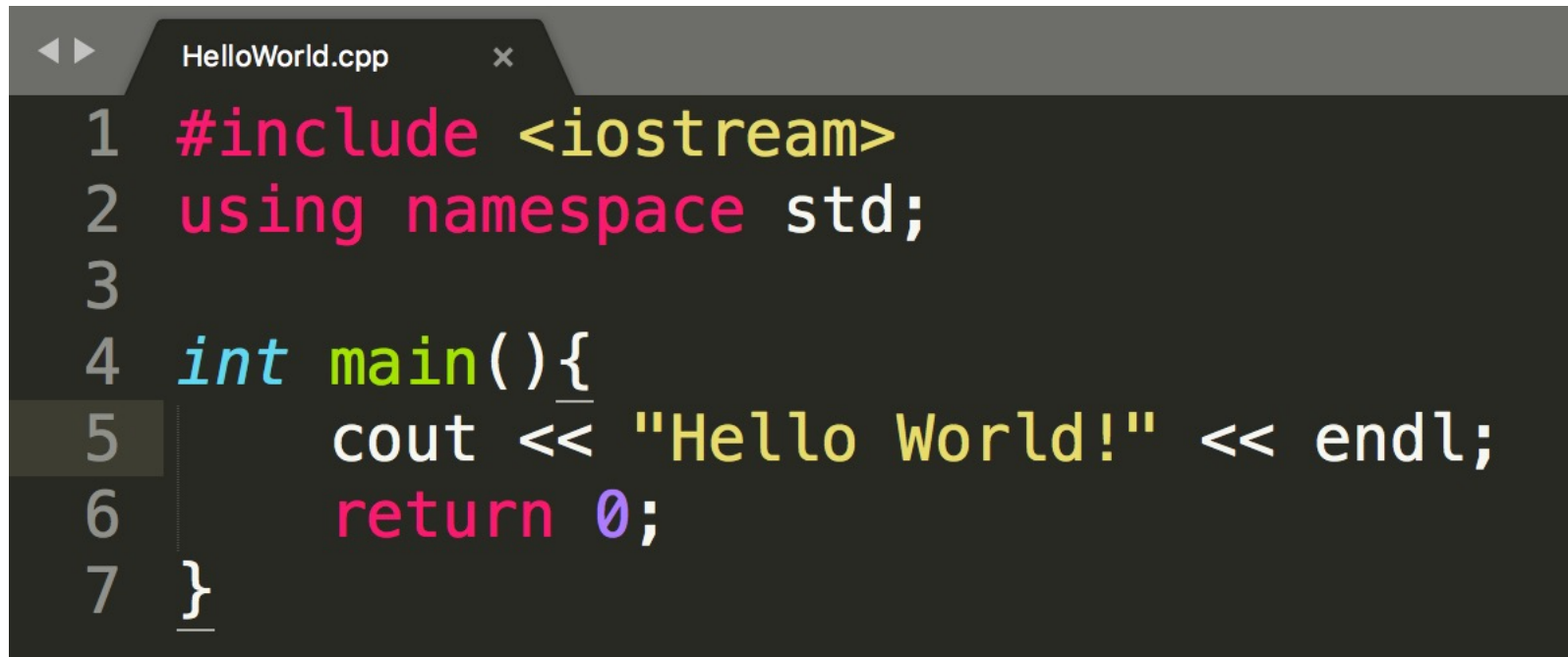
```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     cout << "Hello World!" << endl;
6     return 0;
7 }
```


開發流程



第一個程式：Hello world!

Hello World!

A screenshot of a code editor window titled 'HelloWorld.cpp'. The code is written in C++ and is as follows:

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

The code is color-coded: keywords like 'include', 'using', 'int', 'main', 'cout', 'endl', and 'return' are in pink; standard library names like 'iostream' and 'std' are in yellow; the variable 'int' is in blue; and the function name 'main' is in green. The string "Hello World!" is in yellow. The editor has a dark background and a tab at the top with the file name and a close button.

C++程式結構

特殊符號

字元	名稱	說明
//	雙斜線	為註解的起始符號。
#	井號	為前置處理指令的起始符號。
<>	角括號	當使用 #include 指令時，用來包含要引入的標題檔名。
()	小括號	用於宣告函數，小括號內包含函數參數。
{ }	大括號	用於包含一個或多個敘述。
“ ”	雙引號	用於包含一個字元串列。
;	分號	為敘述的結束符號。

註解

- 「// 欲註解文字」或「/* 欲註解文字 */」
- 被註解的文字不會被編譯
- 一般用來記錄程式碼的含義

插入標題檔

- C++ reference: <http://www.cplusplus.com/reference/>
- 常用標題檔
 - <iostream>：標準輸入與輸出的函數
 - <fstream>：檔案輸入與輸出的函數
 - <cmath>：數學運算函數
 - <ctime>：時間函數
 - <algorithm>：各類演算法
 - <vector>：容器(container)
 - <string>：字串類別
 - <map>：容器(container)

main() 函數

- 程式主體！
- 程式會從main()函數開始執行，main()函數結尾。

```
1 傳回型態 main(參數){  
2      /*  
3      | 程式主體  
4      */  
5  
6      return 傳回值;  
7  }
```

```
1  int main(int argv, char* argc[]){  
2      /*  
3      | 程式主體  
4      */  
5  
6      return 0;  
7  }
```


cout函數

使用方法：

```
cout << 變數或字串1 <<變數或字串2 <<變數或字串3 << ...;
```

- 「<<」
 - 串列輸出運算符（**stream output**），將指定變數的資料或字串常數資料依序向輸出設備移出。

輸出練習

2-01

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      cout << "紅葉晚蕭蕭，長亭酒一瓢。";
6      return 0;
7  }
```

```
[CHENYIs-MacBook-Pro:week1 r07922059$ ./2-01
紅葉晚蕭蕭，長亭酒一瓢。 CHENYIs-MacBook-Pro:week1 r07922059$
```

2-02

```
1  #include <iostream>
2  using namespace std;
3
4  ▼ int main(){
5      cout << "紅葉晚蕭蕭，";
6      cout << "長亭酒一瓢。";
7      return 0;
8  }
```

```
[CHENYIs-MacBook-Pro:week1 r07922059$ ./2-02
紅葉晚蕭蕭，長亭酒一瓢。 CHENYIs-MacBook-Pro:week1 r07922059$
```

2-03

```
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      cout << "紅葉晚蕭蕭，長亭酒一瓢。";
6      cout << "殘雲歸太華，疏雨過中條。";
7      cout << "樹色隨關迴，河聲入海遙。";
8      cout << "帝鄉明日到，猶自夢漁樵。";
9      return 0;
10 }
```

[CHENYIs-MacBook-Pro:week1 r07922059\$./2-03

紅葉晚蕭蕭，長亭酒一瓢。殘雲歸太華，疏雨過中條。樹色隨關迴，河聲入海遙。帝鄉明日到，猶自夢漁樵。 CHENYIs-MacBook-Pro:week1 r07922059\$

endl (end of line)

- Insert newline and flush
 - <http://www.cplusplus.com/reference/ostream/endl/>
- endl是C++標準庫中的操控器（ Manipulator ），包含于<iostream> ，命名空間(namespace)：std 。
 - <https://baike.baidu.com/item/endl/6256824>

\n (換行字元符)

- 「\n」為字串中的換行字元符，當輸出此符號時，會將此符號辨識為換行。
- 特殊字元表

字元	說明
\0	空字元(null space)
\a	響鈴(bell ring)
\b	倒退(backspace)
\t	移到下一個定位(tab)
\n	插入新行(newline)
\f	跳至下一頁起點(form feed)
\r	跳至同一行起點(carriage return)
\"	插入雙引號(double quote)
\'	插入單引號(single quote)
\\	插入反斜線(back slash)

常數與變數

變數 vs. 常數

- 變數(variable)
 - 電腦記憶體中的一個儲存位置。
- 常數(constant)
 - 程式執行中不可改變的資料項目。

宣告變數

資料型態 變數名稱1, 變數名稱2, ...;

- 資料型態
 - 整數、浮點數、字串等。
- 變數名稱
 - 由英文字母、數字、底線符號組合而成，開頭不可為數字。
 - 會區分大小寫。
- 範例
 - `double I_Am_A_Double_Number;`
 - `int _MyInteger;`

宣告常數

```
const 資料型態 變數名稱1, 變數名稱2, ...;
```

- 宣告時要加入 **const** 關鍵字
- 程式進行中，只能被初始化一次！

指定資料

- 「=」：指定運算子
 - 可利用等號給予變數初值，設定初值後也可任意改變數值。
 - 使用「=」變更數值時，指定值的型態與變數型態要相同。

今年是 2022 年 -> 錯 (false)

今天的課程是第二堂課 -> 對 (true)

資料型態

邏輯上的 true/false (真/假)

資料型態	名稱	儲存空間 (位元組數/位元數)	儲存值範圍
bool	布林	1 byte/8bit	0 ~ 255
short int	短整數	2 bytes/16 bit	-32,768 ~ 32767
unsigned short int	短整數(無負號)	2 bytes/16 bit	0 ~ 65,535
int	整數	4 bytes/32 bit	-2,147,483,648 ~ 2,147,483,647
unsigned int	整數(無負號)	4 bytes/32 bit	0 ~ 4,294,967,295
long	長整數	4 bytes/32 bit	-2,147,483,648 ~ 2,147,483,647
unsigned long int	長整數(無負號)	4 bytes/32 bit	0 ~ 4,294,967,295
float	浮點數	4 bytes/32 bit	$\pm 3.4 \times 10^{-38} \sim \pm 3.4 \times 10^{38}$
double	倍精度浮點數	8 byte/64 bit	$\pm 1.7 \times 10^{-308} \sim \pm 1.7 \times 10^{308}$
long double	長倍精度浮點數	12~16 bytes/96~128 bit	$\pm 1.7 \times 10^{-308} \sim \pm 1.7 \times 10^{308}$
char	字元	1 byte/8 bit	0 ~ 255 ASCII碼

character

運算子

序	運算子	說明	結合性
1	::	範圍解析	-
2	++	後置遞增	左
	--	後置遞減	
	()	函數呼叫	
	[]	陣列標註	
	->	指標成員	
3	++	前置遞增	右
	--	前置遞減	
	+	正號	
	-	負號	
	~	位元取反	
	!	邏輯 NOT	
	sizeof	資料的大小	
	&	變數的位址	
	*	指標指向的值	
	new	動態記憶體配置	
	delete	動態記憶體釋放	
	(類型)	強制轉換類型	
4	.* ->*	指標存取	左
5	*	乘	
	/	除	
	%	餘數	
6	+	加	
	-	減	
7	<<	位元左移	左
	>>	位元右移	

序	運算子	說明	結合性
8	<	小於	左
	<=	小於等於	
	>	大於	
	>=	大於等於	
9	==	相等	
	!=	不相等	
10	&	位元 AND	
11	^	位元 XOR	
12		位元 OR	
13	&&	邏輯 AND	
14		邏輯 OR	
15	=	指定	右
	*=	複合指定	
	/=		
	%=		
	+=		
	-=		
	>>=		
	<<=		
	&=		
	^=		
	=		
	? :	條件	
16	,	逗號	左

cin 函數

使用方法：

```
cin >> 變數1 >> 變數2 >> 變數3 >> ...;
```

- 「>>」
 - 串列輸入運算符（**stream input**），將輸入設備的資料儲存至變數中。

<string> 標頭檔

- 提供新的變數型態 **string** 用以儲存字串類型的資料。

練習

- ZeroJudge：從以下題目任選兩題完成，下次上課時檢查。
 - a001：哈囉
 - a002：簡易加法
- Reading: 課本Ch1~Ch3
- 若遇到作業問題，歡迎隨時寄信至：ck1001099@gmail.com