

# C++程式設計基礎

## lesson 4

陳毅

# 時間表

- |         |   |                   |
|---------|---|-------------------|
| • 10/3  | } | • 資料型態、變數、基本輸入輸出  |
| • 10/17 |   | • 流程控制（條件控制、迴圈控制） |
| • 10/31 |   | • 函式與陣列           |
|         |   | • 字元與字串           |
| • 11/14 | } |                   |
| • 11/21 |   | • 指標              |
| • 11/28 |   | • 資料結構            |
| • 12/5  | } |                   |
| • 12/12 | } | • 基礎演算法           |
| • 12/19 |   | • 物件導向程式設計基礎      |
| • 12/26 |   |                   |

lesson 1 - lesson 3 課程內容複習

# 課本題目 – Ch 4

- (6) 寫一C++程式，由鍵盤輸入方程式 $ax^2 + bx + c = 0$ 的 $a, b, c$ ，然後判斷此方程式有一個實根、二個實根、或兩個虛根。
- (10) 寫一C++程式，由鍵盤輸入國文、英文、數學三科的成績，然後計算並顯示總分、平均、與等級，其中平均與等級對應如下表。

平均	等級
90 - 100	A
80 - 89	B
70 - 79	C
60 - 69	D
0 - 59	F

# 課本題目 – Ch 5

- (7) 寫一C++程式，列出攝氏溫度37至39度間隔0.1度的華氏溫度對應表。（華氏 = 攝氏 \* (9/5) + 32）
- (14) 寫一C++程式，重複以cin輸入整數，並加入sum變數中，當輸入為0時，則結束程式。
- (16) 寫一C++程式，利用輾轉相除法求二數的最大公因數(GCD)。

**例** 求 540 和 840 的最大公因數。

**解**

1	540	840	1
	300	540	
4	240	300	1
	240	240	
	0	60	

# 課本題目 – Ch 6

- (5) 寫一C++程式，計算球面積與體積。
  - a) 定義一個sArea(pi, r)函數，接收pi與r參數，傳回球面積給呼叫敘述。
  - b) 定義一個sVolumn(pi, r)函數，接收pi與r參數，傳回球體積給呼叫敘述。
  - c) 在main函數中，呼叫sArea()函數與sVolumn函數，假設球半徑為5, 6, 7, 8, 9, 10。
- (8) 寫一C++程式，以星號(\*)顯示長方形外框或實心長方形。
  - a) 定義一個rectangle(l, w, outline)函數，l與w長方形的長與寬，當outline=1則以星號畫長方形外框，當outline=0則以星號畫實心長方形。
  - b) 從鍵盤輸入三個整數資料l, w與outline，呼叫rectangle函數，畫出長方形。

# 課本題目 – Ch 7

- (4) 寫一C++程式，以亂數模擬擲骰子15000次，並以6個元素的陣列儲存出現1, 2, 3, 4, 5, 6點的次數，結束後輸出1, 2, 3, 4, 5, 6點的次數。
- (6) 寫一C++程式，執行二個二階方陣相加的運算，然後輸出方陣與運算值。
- (7) 寫一C++程式，執行二個三階方陣相乘的運算，然後輸出方陣與運算值。

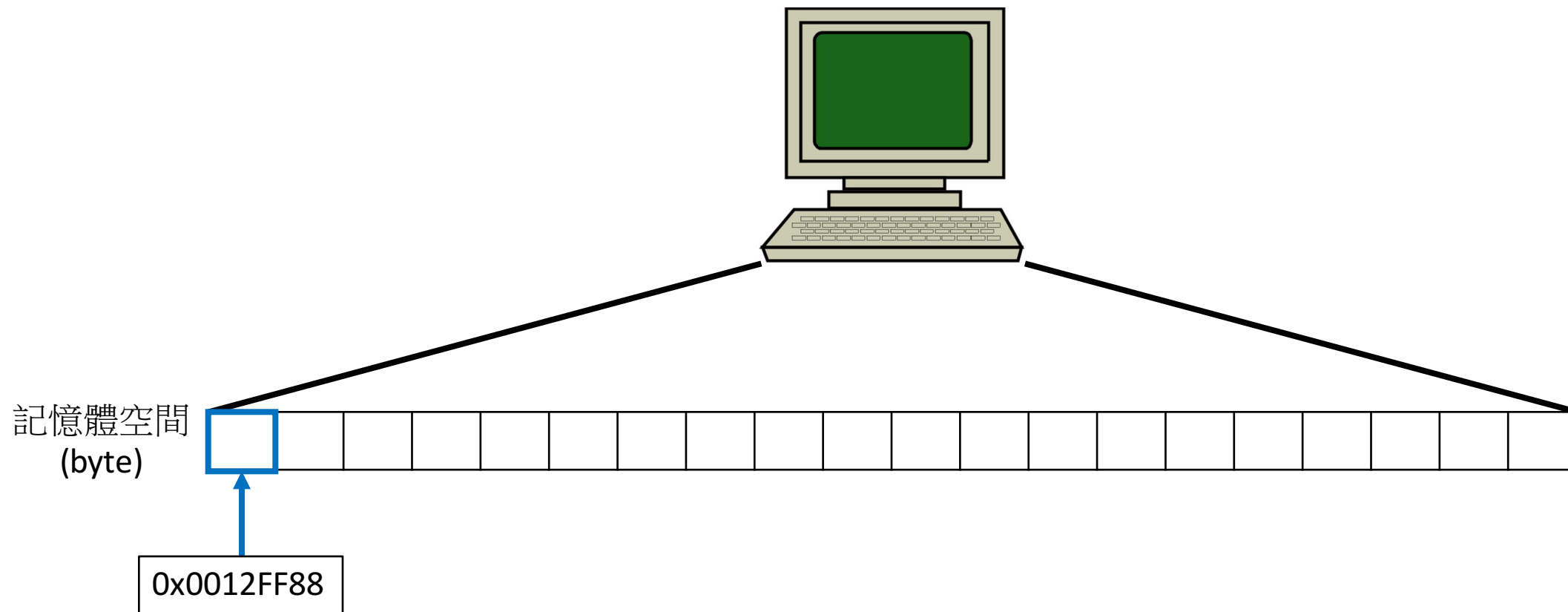
指標



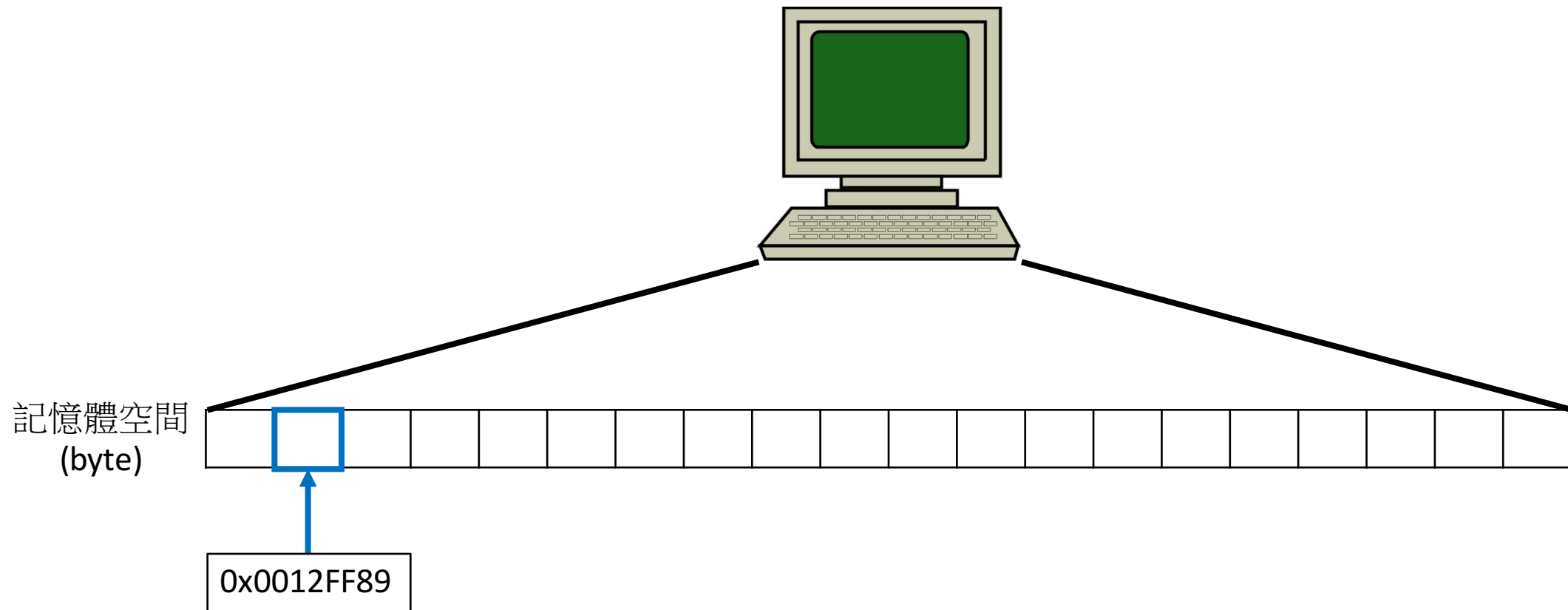
# 今天過後，你要學會...

- 宣告一個用來儲存「**記憶體位址**」的變數，此變數稱作**指標**。
- 使用「位址運算符 **&**」得到一個變數的**記憶體位址**。
- 使用「間接運算符 **\***」得到一個記憶體位址所儲存的**值**。

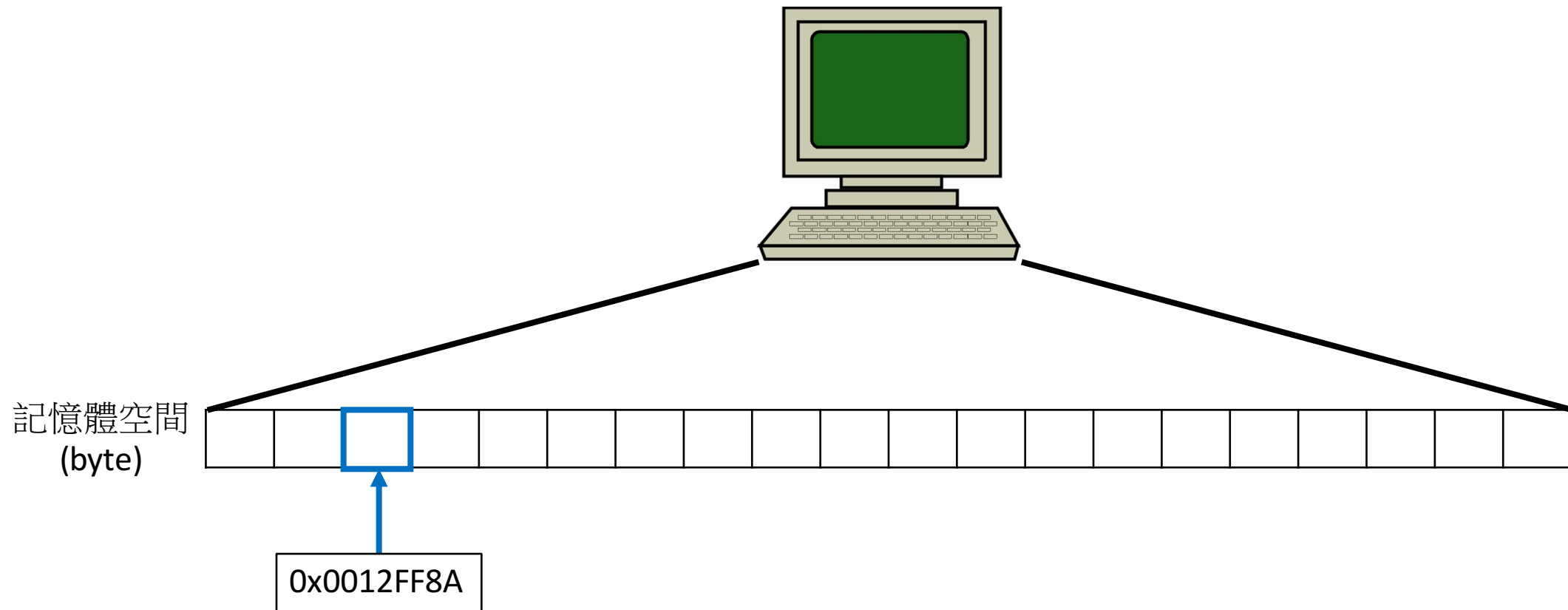
# 電腦的記憶體



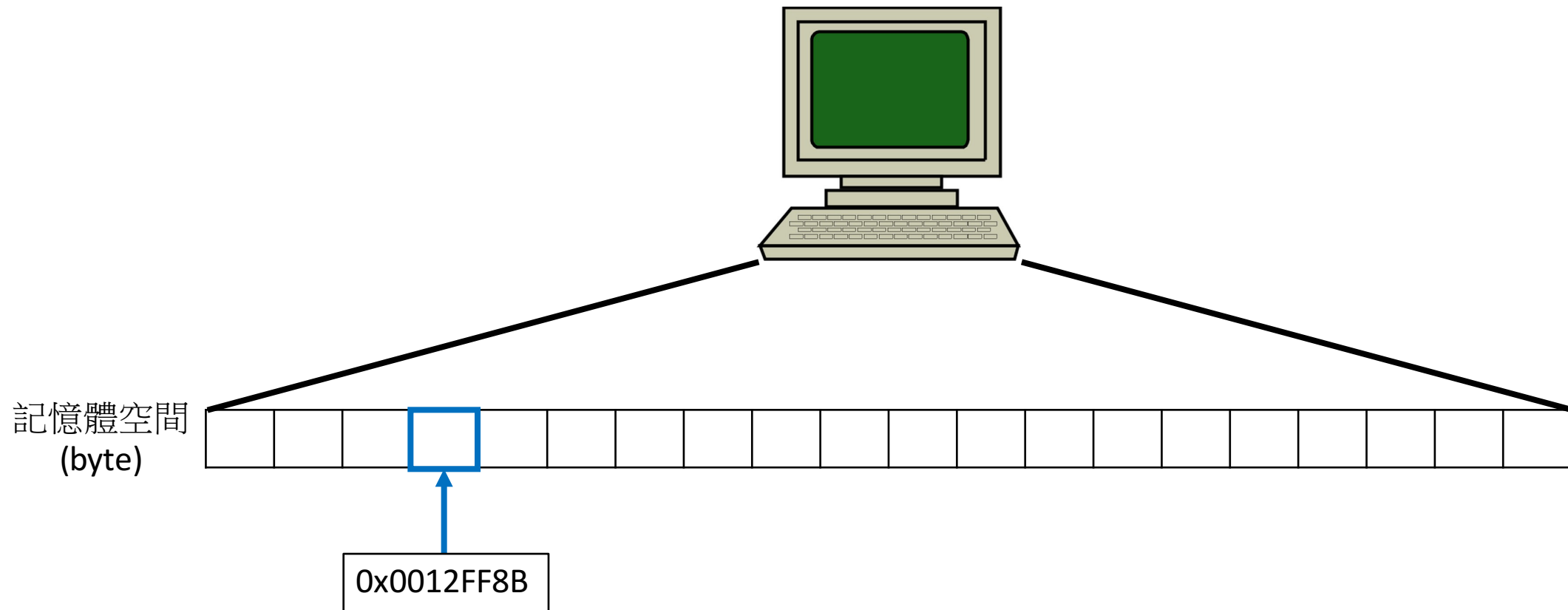
# 電腦的記憶體



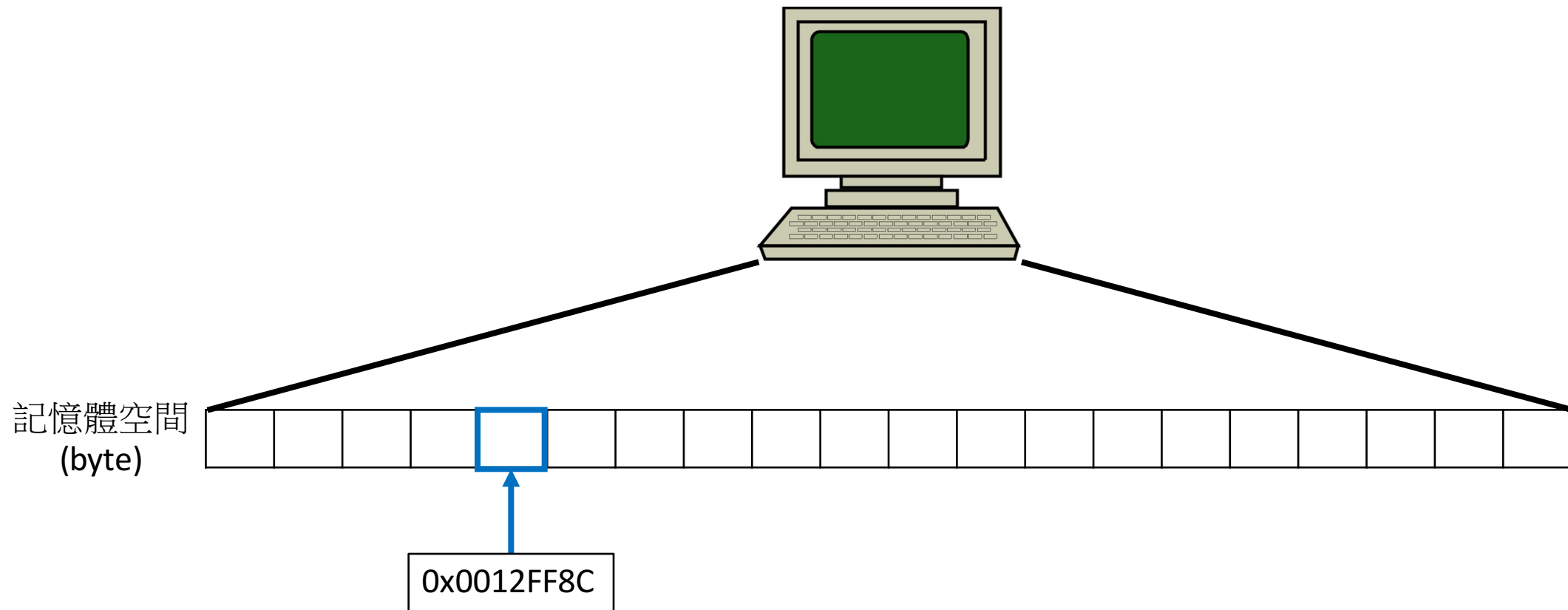
# 電腦的記憶體



# 電腦的記憶體



# 電腦的記憶體



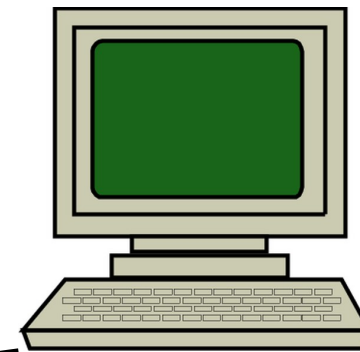
# 複習：sizeof 運算子 (lesson 3)

- 用來計算程式中變數所佔用的記憶體大小，這個運算子在動態配置記憶體時有很大的用處。

```
int a;  
char b;  
float c;  
double d;  
bool e;  
cout << "int: " << sizeof(a) << endl;  
cout << "char: " << sizeof(b) << endl;  
cout << "float: " << sizeof(c) << endl;  
cout << "double: " << sizeof(d) << endl;  
cout << "bool: " << sizeof(e) << endl;
```

# 關於電腦如何儲存一個變數的資料

```
int a = 10;  
double b = 3.14;  
char c = 'A';  
int d = 2000;
```



記憶體空間



名稱：a  
值：10  
型態：int  
長度：4 bytes

名稱：b  
值：3.14  
型態：double  
長度：8 bytes

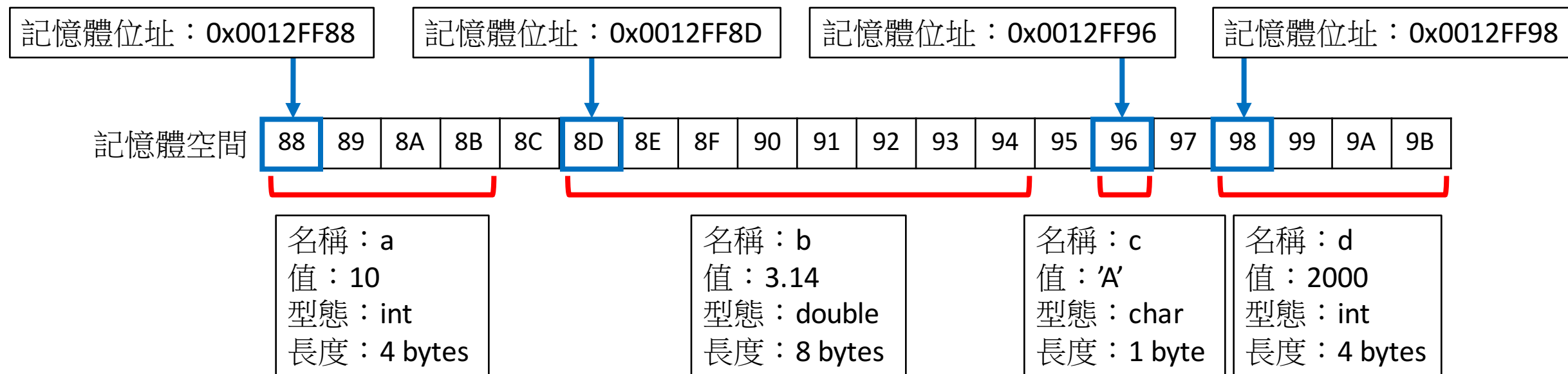
名稱：c  
值：'A'  
型態：char  
長度：1 byte

名稱：d  
值：2000  
型態：int  
長度：4 bytes



# 變數的記憶體位址

- 儲存一個變數的第一個記憶體位址，即為該變數的記憶體位址



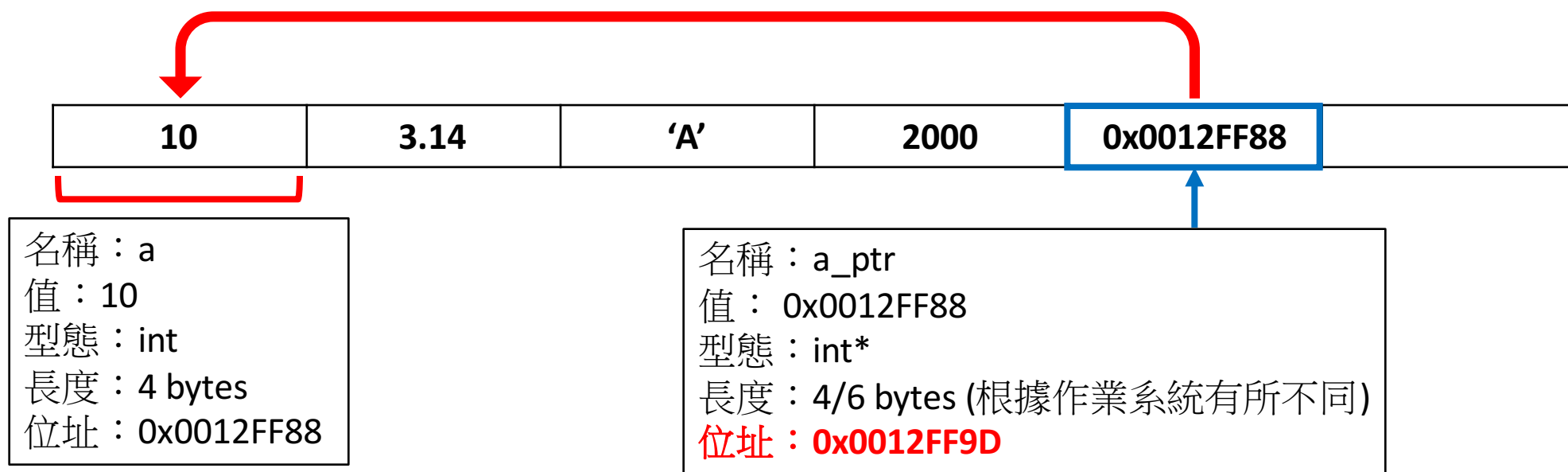
# 知道變數的記憶體位址能做什麼？

- 存取變數的值
  - 可以透過記憶體位址，去存取變數的值。
- 方便傳遞
  - 將一個元素很多的陣列傳遞給函數，要將每個元素都複製一遍傳遞過去。但若知道陣列的記憶體位址，那只要傳遞記憶體位址，函數就可以知道陣列在哪。
- 可用來構築資料結構
  - 如：堆疊(stack)、佇列(queue)、鏈結串列(linked list)、二元樹(binary tree)。

# 指標 (Pointer)

- 指標是一種資料型態，用來儲存記憶體位址。
- 指標本身也具有記憶體空間。

```
int *a_ptr = &a;
```



# 宣告指標

資料型態 \*指標變數;

- 宣告指標變數與宣告一般變數的方法類似，只是在指標變數前面加上「\*」或是在資料型態後面加上「\*」。

容易在同時宣告多個變數時出現誤用

## 誤用

- 宣告兩個指標變數ptr1與ptr2。

```
int* ptr1, ptr2;
```

ptr1為指標變數  
ptr2為int變數

```
int *ptr1, *ptr2;
```

正確

# 指標與變數的參考

- 取得變數的**記憶體位址**
  - 「&」稱作位址運算符(**address-of operator**)，是用來取得變數的位址，也稱作**參考運算符號(reference operator)**。

**&變數名稱**

- 取得一個記憶體位址所儲存的**值**
  - 「\*」稱作間接運算符號(**indirect operator**)，是用來取得參考位址內的值，也稱作**反參考運算符號(de-reference operator)**。

**\*指標名稱**

# 注意事項

- 指標需指向正確的型態

```
float number;  
int *ptr = &number;
```

- 不可用&運算子對常數或運算式取值

```
int *ptr = &3;  
int *ptr2 = &(num1+num2);
```

- 不可在指標並未指向任何記憶體時，使用\*運算子

```
int *ptr;  
cout << *ptr;
```

# 作業

- 從以下題目任選兩題完成，下次上課時檢查。
  - d074：電腦教室
  - d471：0 與 1 的遊戲
  - d066：上學去吧！
  - b981：YoJudge 預練(時間之章)
  - d985：Gran Turismo 5
  - b603：拋物線方程式
  - d563：等值首尾和
- Reading: 課本Ch8.1
- 若遇到作業問題，歡迎隨時寄信至：[ck1001099@gmail.com](mailto:ck1001099@gmail.com)