

【第一講】

C/C++ 程式設計基礎課程簡介

講師: 李根逸 (Ken-Yi Lee), E-mail: feis.tw@gmail.com



課程大綱

- 課程簡介 [P.11]
 - ▶ 課程使用的軟體: Dev C++ [P.12]
 - ▶ 課程網頁: <http://g.csie.org/~kez/c>期數 [P.13]
 - ▶ 課程講義 [P.14]
 - ▶ 參考書籍: C 程式設計藝術 [P.15]
 - ▶ 成績評量 [P.16]
 - ▶ 其他資訊 [P.17]
- 程式語言簡介 [P.19]
- 快速上手 [P.24]
 - ▶ 啟動 Dev C++ [P.25]
 - ▶ 編譯與執行程式 [P.28]
 - ▶ Hello world [P.29]
 - ▶ 記憶體的概念 [P.34]
 - ▶ C / C++ 的運算 [P.35]

課程簡介

- 本課程教導 **C / C++** 程式語言的基礎，適合稍微或沒有程式設計經驗但有興趣的學員。
 - ▶ 但是修習本課程需要有基本的英文打字能力
- 本課程內容包含 **C** 與 **C++** 語言的基礎，在前半段介紹 **C** 語言與 **C++** 共通的部份，後半段補充 **C++** 語言與 **C** 語言的差異（主要是介紹物件導向程式設計的概念）。
 - ▶ 雖然這麼說不盡精確，但可以先把 **C++** 算是 **C** 的一種擴充加強版
 - **【思考】** 那為什麼我們需要兩個語言而不是發明 **C** 語言 2.0？

課程使用的軟體

- 撰寫程式需要安裝具有編譯 **C/C++** 程式能力的整合性開發環境或編譯軟體：
 - ▶ 編譯軟體 (Compiler):
 - Microsoft Visual C++ (MSVC++)
 - The GNU Compiler Collection (GCC)
 - ▶ 整合開發環境 (IDE: 包含某套編譯軟體):
 - Microsoft Visual Studio (MSVC++)
 - Dev C++ (GCC)
 - 其他: Code::blocks, Qt Creator, 等等.
- 教室電腦內裝有 **Microsoft Visual Studio**、**Code::Blocks** 與 **Dev C++**
 - ▶ 上課示範將以 Dev C++ 為主。

課程網頁

- 課程網頁的網址：<http://g.csie.org/~kez/c>期數
- 課程網頁上會放置課程預定進度、每次上課使用的投影片、範例檔和補充教材供學員下載閱讀與練習。如果課程臨時有什麼變動，我們會更新網頁通知。
 - ▶ 因天氣因素停止上課的標準以人事行政局為準
- 因為課程進度將隨著上課情形與同學反應而有所變動，所以我們的課程網頁將會動態更新。
 - ▶ 網頁內容如果有任何缺失或資料遺漏還麻煩大家提醒！
- **Facebook 討論區**：<http://goo.gl/6GpVf>
 - ▶ 提供訊息發佈、問題發問或資訊交流的管道

課程講義

- 上課用的投影片紙本會作為上課用的講義
 - 講義為講師個人學習、實務與教學經驗的集成，雖然已經不停的更新補充，但是編寫時錯誤缺漏或不盡如人意的地方難免，有任何問題或建議請跟講師反應。
- 講義內容主要包含四個部分：觀念、範例、補充與習題
 - ▶ 範例：
 - 我們上課會細講某些選過的範例，而未講述的範例請回家自行練習。範例解答在課程網頁會提供，請盡量做到能夠不參考解答下完全獨力寫出範例。初學者請每隔一段時間重複練習範例到熟練。
 - ▶ 補充：
 - 上課不會細講的內容，通常是已經學過要複習時看。
 - ▶ 習題：
 - 我們不提供完整的習題解答也不建議每個人都要寫習題，尤其初學者請以範例為主。對於習題有任何問題請直接詢問講師，講師會依照學生的反應給予適當提示或示範。

參考書籍

- 本課程主要參考書籍：
 - ▶ C : How to Program 《C 程式設計藝術》
- 其他課程相關 **C** 參考書籍有：
 - ▶ The C Programming Language
 - ▶ C Programming: A Modern Approach
 - ▶ Headfirst C
- 其他課程相關 **C++** 參考書籍有：
 - ▶ C++ How to Program
 - ▶ The C++ Programming Language
- 其他中文參考書籍：
 - ▶ C 語言教學手冊 〈旗標〉
 - ▶ C/C++ 無痛學習教本 〈旗標〉
- 建議初學程式語言者還是需要一本入門參考書籍

成績評量

■ 出席率與隨堂練習 (40%)

70 分及格

■ 期中與期末測驗 (各 30%)

第 1 堂課

第 2-4, 6-8, 10-12 堂課

第 5, 9 堂課

時		時		時	
19:00	上課	19:00	隨堂練習	19:00	測驗開始
20:00	休息	19:20	批改與講解	20:50	測驗結束
20:10	上課	19:30	休息與 Q&A	21:00	檢討與 Q & A
	隨堂練習	19:40	上課	21:30	下課
21:00	休息 Q&A	20:40	休息		
21:30	下課	20:40	上課		
		21:30	下課		

其他資訊

■ 門禁系統：

- ▶ 假日整天或平日的晚上大門會上鎖，我們會發放感應式的門禁卡讓學員使用 (需100元押金)。於最後一次上課結束時需要繳回卡並拿回押金。請注意門禁卡只能使用在系館正門。

■ 垃圾處理：

- ▶ 請千萬不要將垃圾丟棄至廁所內垃圾桶！
- ▶ 請拿至一樓中庭 (電梯旁小門外) 垃圾分類區丟棄

■ 在校內停車費用：

- ▶ 可以使用學員證或請講師在停車單據上加蓋本班班章會有等同校內學生停車的折扣

■ 學員證與收據

- ▶ 學員證約在第一次上課發放
- ▶ 收據約在第二次上課發放
- ▶ 錯過同學請記得在上課期間跟老師索取

程式語言簡介

程式語言

■ 程式語言 (Programming Language) :

"A *programming language* is a *formal language* designed to communicate instructions to a machine, particularly a computer.

Programming languages can be used to create programs that **control** the **behavior** of a **machine** and/or to express algorithms precisely." - from *Wikipedia*

- ▶ 程式語言 (programming language) 是用來控制 (control) 機器 (machine) 行為 (behavior) 的形式 (formal) 語言

什麼是機器 / 電腦 ？

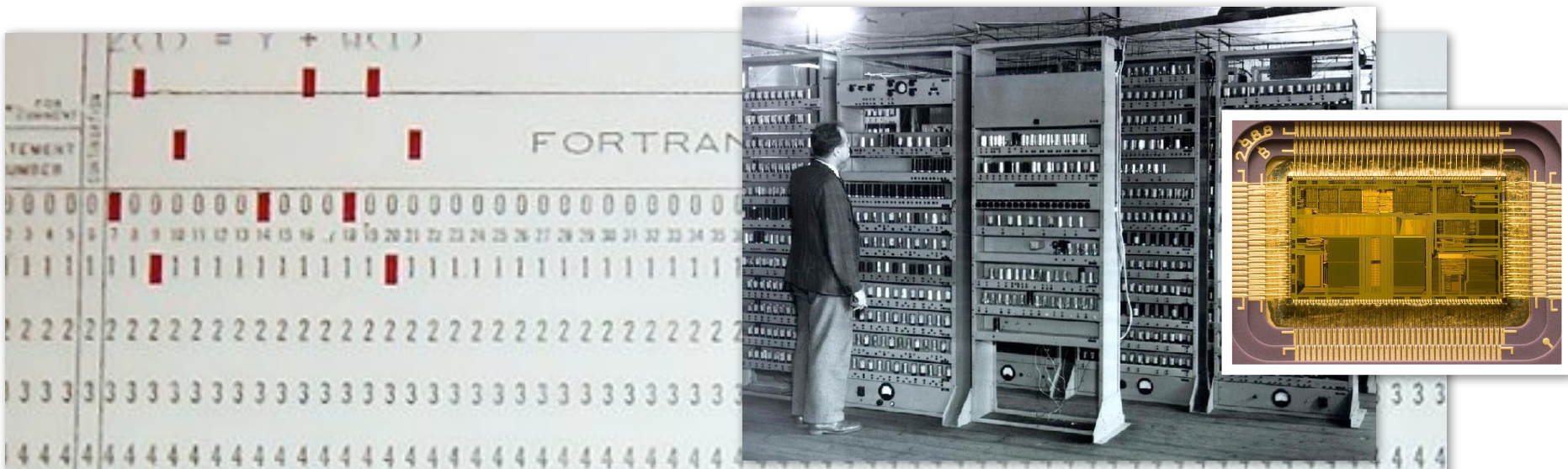


from *Wikipedia*

電腦的組成

- 抽象來說，電腦內包含了：
 - ▶ 輸入單元 (Input): 例如鍵盤和滑鼠 (感官)
 - ▶ 輸出單元 (Output): 例如螢幕和印表機 (嘴巴、肢體)
 - ▶ 輔助儲存單元 (Storage): 例如硬碟和光碟
 - ▶ 記憶單元 (Memory): 例如記憶體
 - ▶ 中央處理單元 (CPU)

程式設計核心
(記憶與計算)



程式語言的分階

■ 程式語言一般來說可分為三個階層：

▶ 機器語言：

- 01001101011111010001010000010110
- 01010011011110110101110000001011
- 010001111000101010111101001101011

▶ 組合語言：

- LOAD A
- ADD B
- STORE SUM

指令集: 硬體支援的指令 (x86, ARM, ...)

▶ 高階語言：

- SUM = A + B

常見的高階語言有 C, C++, Java, Pascal, Python, ...

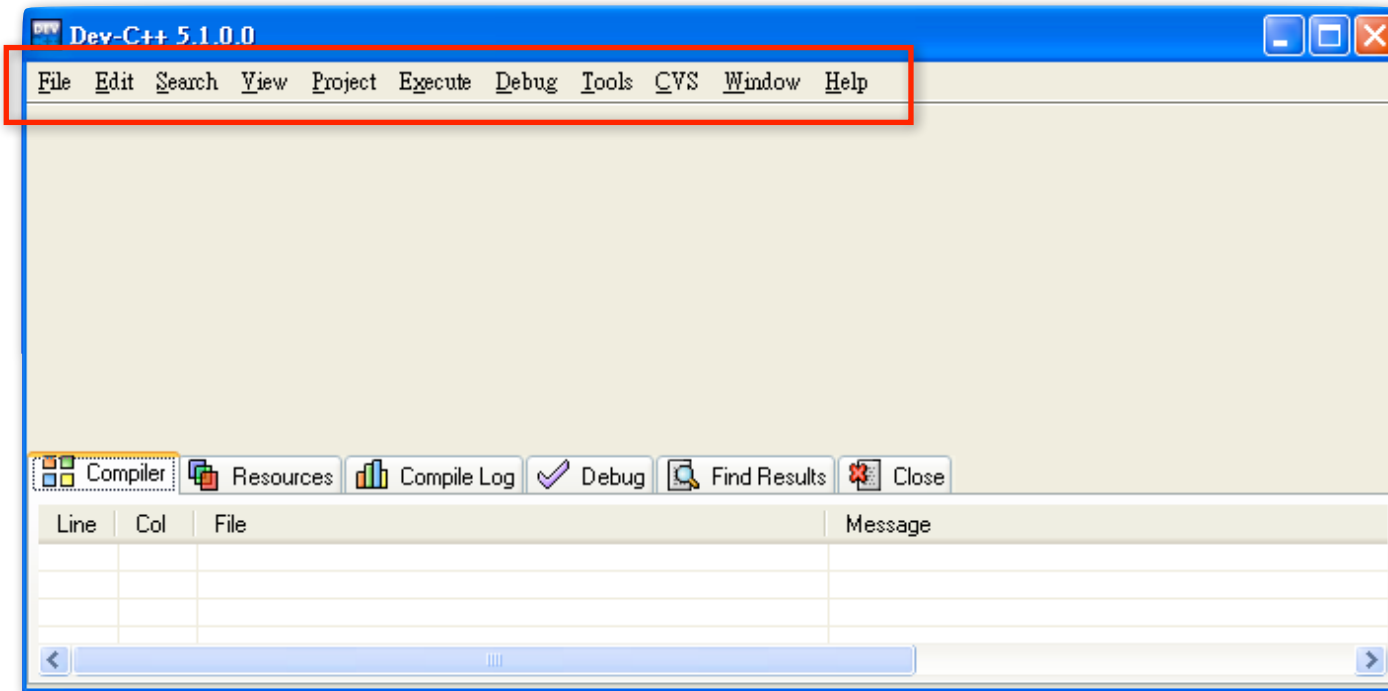
C / C++ 語言的歷史

- **C 語言是一種高階語言**
 - ▶ 西元 1972 年由 Dennis Ritchie 所開發。
 - ▶ 西元 1989 年，美國國家標準局通過 ANSI X3.159-1989 (俗稱 C89) [目前常用]
 - ▶ 西元 1990 年，國際標準組織 (ISO) 通過 ISO/IEC 9899:1990 (俗稱 C90) [內容與 C89 雷同]
 - ▶ 西元 1999 年，國際標準組織 (ISO) 通過 ISO/IEC 9899:1999 (俗稱 C99)
- **C++ 語言原則上包含了 C 語言**
 - ▶ 西元 1998 年，國際標準組織 (ISO) 通過 ISO/IEC 14882:1998 (俗稱 C++98) [目前常用]
- **C++ 加入了一些傳統 C 語言沒有的『語言特性』，最為人熟知的就是『物件導向程式設計 (OOP)』的相關語法**

快速上手

啟動 Dev C++

- 「開始 > 所有程式 > Bloodshed Dev C++ > Dev C++」



整合開發環境 (IDE)



Dev C++ 的縮排設定

- 一般像是 **Dev C++** 和 **Microsoft Visual Studio** 這類的整合開發環境 (**IDE**)，可以自動判斷程式內容幫助你縮排。因此在程式碼的撰寫過程中，軟體會自動加入 **Tab** 或空白。以 **Dev C++** 為例，可以到這裡修改縮排設定：
 - ▶ 「Tools (工具) > Editor Options (編輯器選項) > General (一般)」
- 此外，**Dev C++** 預設會自動在輸入左括號後加上右括號，不喜歡的話可以到下面的設定修改：
 - ▶ 「Tools (工具) > Editor Options (編輯器選項) > Code Insertion (插入程式碼) > Symbol Completion (符號完成)」把「Enable symbol completion (啟用符號完成)」關閉

基本操作

■ 開新檔案：

- ▶ 「File (檔案) > New (開新檔案) > Source file (原始碼)」

■ 儲存檔案：

- ▶ 「File (檔案) > Save [As] (儲存)」
 - 檔名請勿含有空白並避免含有特殊符號 (例如中文)
 - 使用 `cpp` 作為副檔名

■ 開啟一個已經存在的檔案：

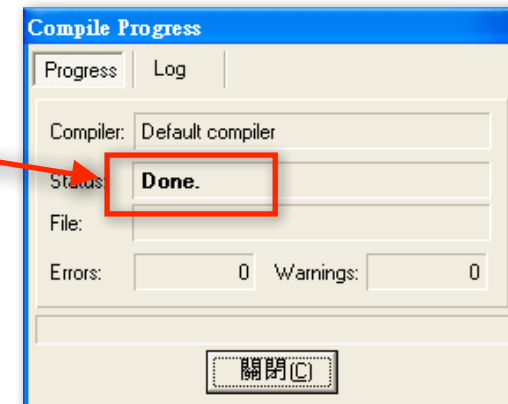
- ▶ 「File (檔案) > Open (開啟)」

編譯與執行程式

- 程式檔 (.c, .cpp) 需要經由編譯 (Compile) 的動作才能成為執行檔 (.exe) 讓電腦執行 (Run)
 - ▶ 「Execute (執行) > Compile (編譯)」 《F9》
 - 編譯成功後會產生執行檔

『Done』表示編譯成功

可以用《F11》進行編譯後直接執行



- ▶ 「Execute (執行) > Run (執行)」 《F10》
 - 要先編譯成功才會有執行檔可以執行
- 常用的指令：編譯並執行 《F11》
 - ▶ 「Execute (執行) > Compile&Run(編譯並執行) 《F11》

【範例】 Hello world [1]

- 請開啟一個新檔案並輸入下列程式碼後編譯並執行《F11》：

```
/* My first program */  
#include <stdio.h>  
  
int main() {  
    printf("Hello world !\n");  
    return 0;  
}
```

`printf`（“字串”）是一個可以用來印出字串在螢幕上的函式

試著在電腦裡找到你的程式碼檔案跟執行檔後點擊執行看看

【範例】 `helloworld_1.cpp`

【範例】 Hello world [2]

- 請寫一程式，輸入下列程式碼後編譯並執行 **《F11》**：

```
/* My first program */  
#include <stdio.h>  
#include <stdlib.h>  
  
int main() {  
    printf("Hello world !\n");  
    system("pause");  
    return 0;  
}
```

`system("指令")` 是一個在作業系統上執行指令的函式

`pause` 是 MS-DOS 作業系統中用來暫停並等待輸入的指令

【範例】 `helloworld_2.cpp`

【範例】 Hello world [3]

- 請寫一程式，輸入下列程式碼後編譯並執行 «**F11**»
：

```
/* My first program */  
#include <stdio.h>  
#include <stdlib.h>  
  
int main() {  
    printf("Hello ");  
    printf("world !\n");  
    system("pause");  
    return 0;  
}
```

【範例】顯示文字

- 請寫一個程式顯示下面的文字圖案給使用者看

```
H      H      A
H      H      A A
HHHHHHH  A      A
H      H  AAAAAAA
H      H  A      A
```

- ▶ 【思考】 我們需要使用幾次 `printf(...)` ?

【補充】跳脫串列

- 在 `printf` 的格式字串中，我們需要使用跳脫串列來讓電腦顯示一些無法或不易直接輸入的符號或文字：

跳脫串列	說明
<code>\n</code>	Newline (換行)
<code>\t</code>	Tab (跳位)
<code>\\</code>	反斜線
<code>\"</code>	雙引號
<code>\b</code>	發出逼逼聲

【範例】 計算兩數相加

- 試寫一程式如下：

```
/* Addition program */
#include <stdio.h>
#include <stdlib.h>
int main(){
    int integer1;
    int integer2;
    int sum;
    printf("Please enter the first integer: ");
    scanf("%d", &integer1);
    printf("Please enter the second integer: ");
    scanf("%d", &integer2);
    sum = integer1 + integer2;
    printf("Sum is %d\n", sum);
    system("pause");
    return 0;
}
```

【範例】 add.cpp

記憶體的概念

```

/* Addition program */
#include <stdio.h>
#include <stdlib.h>
int main() {
    int integer1;
    int integer2;
    int sum;
    printf("Please enter the first integer: ");
    scanf("%d", &integer1);
    printf("Please enter the second integer: ");
    scanf("%d", &integer2);
    sum = integer1 + integer2;
    printf("Sum is %d\n", sum);
    system("pause");
    return 0;
}

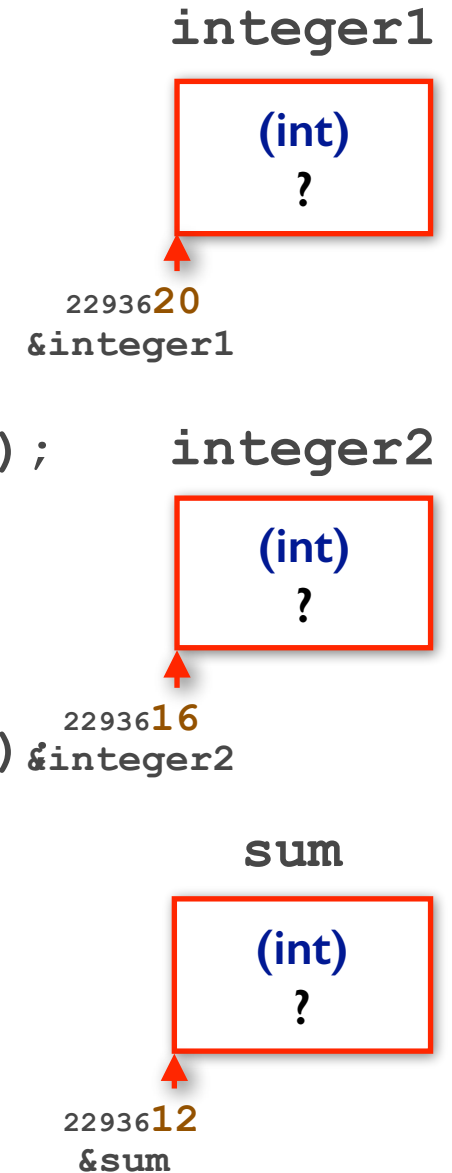
```

int 表示整數型別

scanf 函式會從鍵盤讀入資料後存放在指定的變數 (%d 表示讀入一個整數)

printf 內 %d 表示印出一個整數

【思考】 printf("Sum is sum\n"); 不行嗎?



C / C++ 的運算

■ 3 + 4 * 5 是多少？

- ▶ 運算時請注意運算子優先順序！
 - 先乘除後加減
- ▶ 可以適當加上括號 () 來調整優先順序！

■ 賦值運算子 (=)

- ▶ 『賦值』是指將等號左邊的變數賦予等號右邊的值
 - 一般情況下只有賦值運算子可以改變變數的值

運算	運算子
加法	+
減法	-
乘法	*
除法	/
餘數	%
賦值	=

【範例】 計算乘積

- 請寫一個程式讓使用者依序輸入兩個整數後算出兩個整數的乘積顯示給使用者看

- ▶ 提示：請由 `add.cpp` 開始改起

```
Please enter the first integer: 3
Please enter the second integer: 4
Product is 12
```

【範例】 `multi2.cpp`

- 請寫一個程式讓使用者依序輸入三個整數後算出三個整數的乘積顯示給使用者看

- ▶ 提示：請由 `multi2.cpp` 開始改起

```
Please enter the first integer: 3
Please enter the second integer: 4
Please enter the third integer: 5
Product is 60
```

【範例】 `multi3.cpp`

【範例】 更多計算

- 請寫一個程式讓使用者分別輸入三個整數後，算出三個整數的和、平均值、乘積並顯示給使用者看

```
Please enter the first integer: 3
Please enter the second integer: 4
Please enter the third integer: 5
Sum is 12
Average is 4
Product is 60
```

【範例】 `calc.cpp`

- 請寫一個程式讓使用者輸入一個整數後顯示該數的平方數值

```
Please enter the integer: 3
9
```

【範例】 `squared.cpp`

printf 與 scanf 函式

- `printf` 可以連續輸出多個變數：

```
printf("Sum is %d\n", sum);  
printf("Average is %d\n", average);  
printf("Product is %d\n", product);
```

可連在一起寫成：

```
printf("Sum is %d\nAverage is %d\nProduct is %d\n",  
sum, average, product);
```

- `scanf` 可以連續讀取多個變數：

```
scanf("%d%d%d", &integer1, &integer2, &integer3);
```

- `printf` 可以在 `%` 後加上最少顯示位數：

```
printf("Sum is %5d\n", sum);
```

留五格

```
printf("Sum is %05d\n", sum);
```

留五格，不足補零

【範例】位數解析

- 試寫一程式讓使用者輸入一整數後，顯示其個位數字
 - ▶ 提示：`%` 運算子可以用來求餘數

範例輸入：12345
範例輸出：5

範例輸入：54321
範例輸出：1

【範例】 `digit1.cpp`

- 試寫一程式讓使用者輸入一整數後，顯示其百位數字

範例輸入：12345
範例輸出：3

範例輸入：54321
範例輸出：3

【範例】 `digit3.cpp`

初學 C / C++ 程式設計流程

- 步驟一 [空殼]：
 - ▶ 寫出 main
- 步驟二 [流程]：
 - ▶ 依照執行順序寫出輸入與輸出的程式碼，對於未知或未定的數值給予一個變數名稱。
- 步驟三 [資料]：
 - ▶ 檢查所有的變數在程式碼中是否在正確的時候給予正確的值

初學寫程式碼時在乎的是？

■ 正確性

- ▶ 程式碼是否正確的產生想要的結果

■ 可讀性

- ▶ 程式碼是否容易閱讀與理解

■ 效率

- ▶ CPU 使用效率 (時間效率)

- 是否使用最少的指令

- ▶ 記憶體使用效率 (空間效率)

- 是否配置了最少的記憶體空間

■ 彈性

- ▶ 是否容易依照需求的變更改變程式碼

習題 [1]

- **[E0101]** 試寫一程式印出下列文字：

Alice: How are you ?

Bob: Fine. Thank you.

- **[E0102]** 試寫一程式輸出下列文字

Welcome to "C" world !

Hope you have a nice day !

▶ 提示：利用輸入 \" 來顯示雙引號

- **[E0103]** 試寫一程式，輸入一個整數後，印出該整數加 7 後的值

- **[E0104]** 試寫一程式印出 **129263 + 54628** 的結果

習題 [2]

- **[E0105]** 試寫一程式讓使用者輸入一個長方形的長跟寬後，顯示該長方形的周長與面積
- **[E0106]** 試寫一程式，讓使用者輸入一個整數 x 後，計算出 $3x^2+4x-1$ 的值
- **[E0107]** 試將 `helloworld_2.cpp` 改寫成程式碼內容只有三行但執行結果也一樣的程式碼
- **[E0108]** 試寫一程式，輸入西元年份後，印出民國年份
 - ▶ 民國年份 = 西元年份 - 1911

習題 [3]

- **[E0109]** 試寫一程式，輸入華氏溫度後，印出攝氏溫度的值
 - ▶ 攝氏溫度 = $5 \times (\text{華氏溫度} - 32) / 9$
- **[E0110]** 試寫一程式，讓使用者輸入自己的身高與體重後，顯示使用者的身體質量指數 (**BMI**) 無條件捨去至小數點後第二位：
 - ▶ $\text{BMI} = \text{體重 (kg)} / \text{身高平方 (m}^2\text{)}$
- **[E0111]*** 試寫一程式印出 **129263 * 54628** 的結果
- **[E0112]** 試寫一個程式，讓使用者輸入兩個整數後，計算出兩個整數的平均值至小數點後二位





