



巨匠線上真人

# C 語言程式設計

[www.pcschoolonline.com.tw](http://www.pcschoolonline.com.tw)

## 同學，歡迎你參加本課程

- ☑ 請關閉你的FB、Line等溝通工具，以免影響你上課。
- ☑ 考量頻寬、雜音，請預設關閉攝影機、麥克風，若有需要再打開。
- ☑ 隨時準備好，老師會呼叫你的名字進行互動，鼓勵用麥克風提問。
- ☑ 如果有緊急事情，你必需離開線上教室，請用聊天室私訊給老師，以免老師癡癡呼喚你的名字。
- ☑ 軟體安裝請在上課前安裝完成，未完成的同學，請盡快進行安裝。

# 課程檔案下載

The screenshot displays the homepage of the Juei Computer Online Real Person Course website. The header features the site's name and navigation links. A main banner promotes Python learning. A dropdown menu is open, highlighting the 'Course Archive Download' option. An orange callout bubble points to this option.

巨匠電腦線上真人 開課查詢 免費體驗專區 課程總覽 專業師資 學員專區 講師專區 最新消息

360 f YouTube

您好! 登出

點數卡產品兌換  
APCS檢測專區  
公告專區  
我的課表  
IT真人課程劃位  
電腦分校課程劃位  
外語真人課程劃位  
美語分校課程劃位  
取消劃位  
**課程檔案下載**  
上課權益查詢  
教學平台測試  
學習諮詢  
常見問題  
個資維護  
忘記密碼  
登出

程式語言 **好難學?**  
那是因為  
你還沒學過 **Python!**  
(線上老師 **LIVE** 直播教學 · 搶先看)

巨匠電腦真人課程

# ZOOM 學員操作說明

The screenshot shows the Zoom interface with several key elements highlighted and numbered:

- 5 查看選項/共同註記/筆 (連連看)**: A dropdown menu is open from the '共同註記' (Annotate) button in the top toolbar, showing options: '原始大小' (Original Size), '請求遠端控制' (Request Remote Control), '共同註記' (Annotate), and '退出全螢幕' (Exit Full Screen). The '共同註記' button in the toolbar is also highlighted with an orange box.
- 2 共享螢幕 (指導演練; 點評作品)**: A callout box points to the '共享螢幕' (Share Screen) button in the bottom toolbar. The callout text reads: '老師須先停止共享螢幕 才能請學生共享螢幕' (The teacher must first stop sharing the screen before asking the student to share the screen).
- 1 聊天**: A callout box points to the '聊天' (Chat) button in the bottom toolbar.
- 3 與會者/舉手**: A callout box points to the '與會者' (Participants) button in the bottom toolbar. A secondary callout box points to the '舉手' (Raise Hand) button within the '與會者' window.
- 4 解除靜音**: A callout box points to the '解除靜音' (Unmute) button in the bottom toolbar.

The '與會者' (Participants) window shows a list of participants:

- 張 張齡月 (我)
- 許 婷婷

The '舉手' (Raise Hand) button is highlighted with an orange box.

# 本課程各堂教學主題

成品

◆ 第一堂：Dev-C++ 開發C語言

成品

◆ 第二堂：型式、變數、運算子及運算式

成品

◆ 第三堂：決策控制句型

成品

◆ 第四堂：重複處理句型

成品

◆ 第五堂：技術主題 1 · 陣列應用

成品

◆ 第六堂：技術主題 2 · 專屬型式



C 語言程式設計

# 第一堂：Dev-C++ 開發C語言

# 🔑 本次課堂 · 教學重點 🔑

- 1-1. 工具起源與線上發展史
- 1-2. 工具安裝、設定、操作
- 1-3. 設計新手學前觀察重點
- 1-4. 線上註解新增與移除化
- 1-5. 引用標題化 【 #include 】
- 1-6. 執行主函數 【 main 】
- 1-7. 格式化存取 【 scanf . printf 】
- 1-8. 程式設計、編譯、執行
- 1-9. 範例實作研討





# \*研討節次.

## 1-1. 工具起源與線上發展史



「**Dev-C++工具**」提供「**整合式開發環境・IDE**」進程式開發，設計人員可線上選用「**C、C++**」程式語言來進行「**系統裝置**」功能控制設計，用於開發支援「**MicroSoft Windows**」平台應用程式，同時也是語法研習者最喜愛的熱門工具選項。其「**工具IDE**」是利用「**Delphi**」程式語言所開發出來的，其使用「**MinGW、GDB**」作為編譯系統與除錯系統依據，並搭配「**GCC MinGW**」或「**TDM-GCC**」的「**64位元版本**」作為它的編譯器，也可以使用「**Cygwin**」或任何其他基於「**GCC編譯器**」組合使用，目前線上「**衍生版本 5.x**」可架設在「**64位元版本・Windows 8.10**」及「**32位元版本・Windows 7**」兩種作業系統下進行「**指定編譯化**」版本選取，同步開發設計功能強大。

「**Dev-C++工具**」歷經三個創展時期階段，除了「**原創版本・Dev-C++**」外，還有「**擴展版本・wxDev-C++**」，以及「**衍生版本・Orwell Dev-C++**」。巨匠小幫手將重點內容整理於下列表格，想要掌握更多相關重點資訊者，請上網「**維基百科**」，<https://zh.wikipedia.org/wiki/Dev-C++> 頁面瀏覽查閱。


創展順序	作者資訊	代表版本	起始時期
Dev-C++	Bloodshed Software	4.0	2000.09.10
wxDev-C++		5 Beta	2002.03.19
Orwell Dev-C++	Orwell	5.x	2011.08.27

# \*研討節次.


1-2. 工具安裝、設定、使用

請依教學人員示範與指導動作來完成下載「Dev-C++」指定版本並進行線上安裝。如果無法參與課堂研習者，請參考「[附錄圖片](#)」步驟來自行完成。上課前請確認「[開發工具](#)」已正式安裝完畢，再進行線上相關設計應用！

---

(1) 開啟瀏覽器，下載、安裝開發工具 |  參考附錄圖片( [A1](#) 至 [A4](#) )

- 開啟「[Google Chrome](#)」輸入「[dev c++ 5.11](#)」，進入搜尋載點頁面【[阿榮福利味](#)】
- 依個人需求考量，選擇「[裝置版本](#)」後，執行後續動作【[下載、解壓縮](#)】

(2) 進入開發工具後，進行相關個人化設定 |  參考附錄圖片( [B1](#) 至 [B2](#) )

- 選單語言 ( [Language](#) ) - 英文 / 預設 English(Original)  
- 中文 / 自訂 [Chinese\(TW\)](#)
- 編輯字型 ( [Font](#) ) - 英文 / 預設 Consolas  
- 中文 / 自訂 [偏好字型及相關格式](#)

PS. 巨匠小幫手建議將[字型](#)重新設定成【[細明體](#) / [標楷體](#) · [12點](#) / [14點](#) / [16點](#)】等寬字型大小。

---

## 🔍 參考附錄圖片(A1)



動作 1 - 輸入【工具代號】



動作 2 - 點選【連結選項】

## ④ 參考附錄圖片(A2)



動作 3 - 下移【作業頁面】



動作 4 - 點選【第一載點】

## 🔗 參考附錄圖片(A3)



動作 5 - 點選【第二載點】



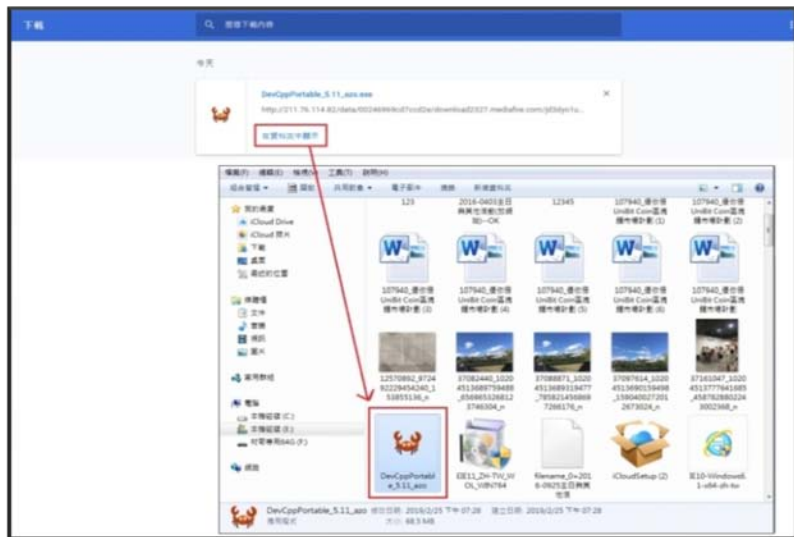
動作 6 - 點選【最終載點】



## 參考附錄圖片(A4)



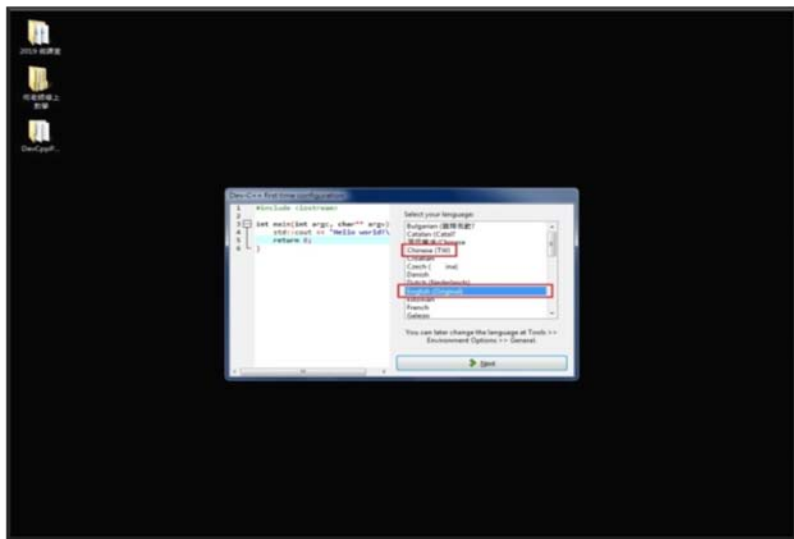
## 動作 7 - 進行【線上下載】



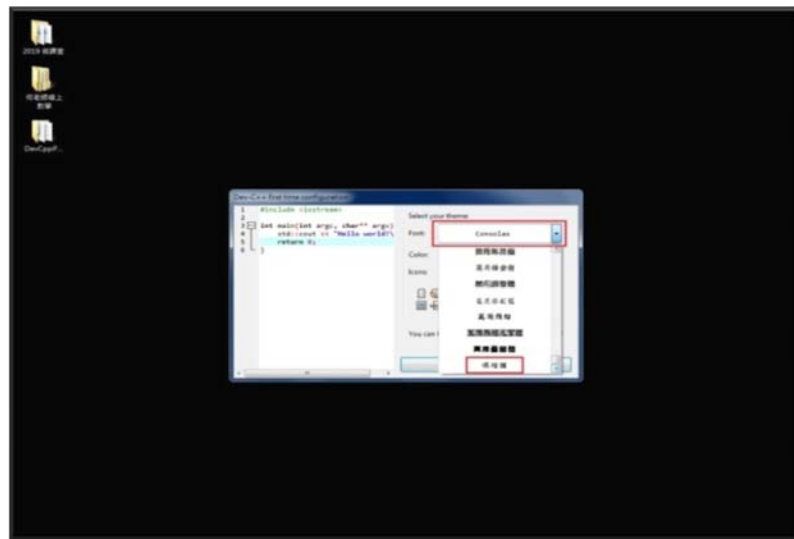
## 動作 8 - 雙擊【安裝程式】



## 🔗 參考附錄圖片(B1)

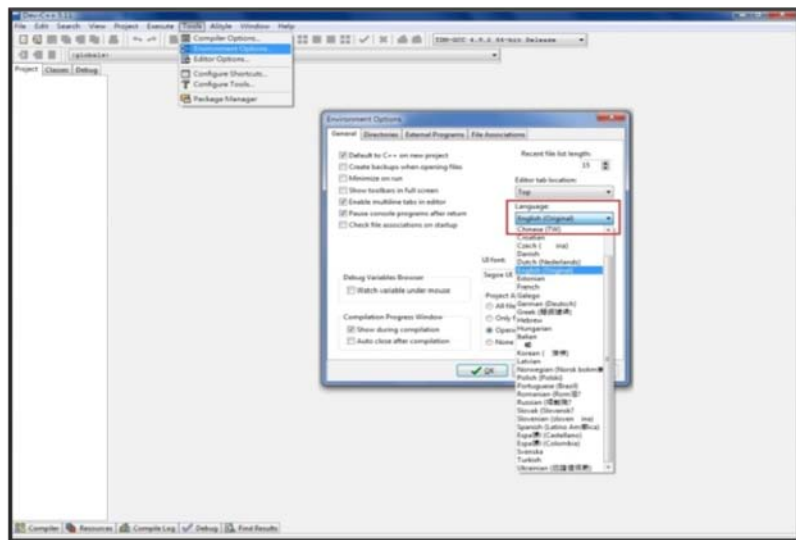


動作 1 - 初訂【選單語言】

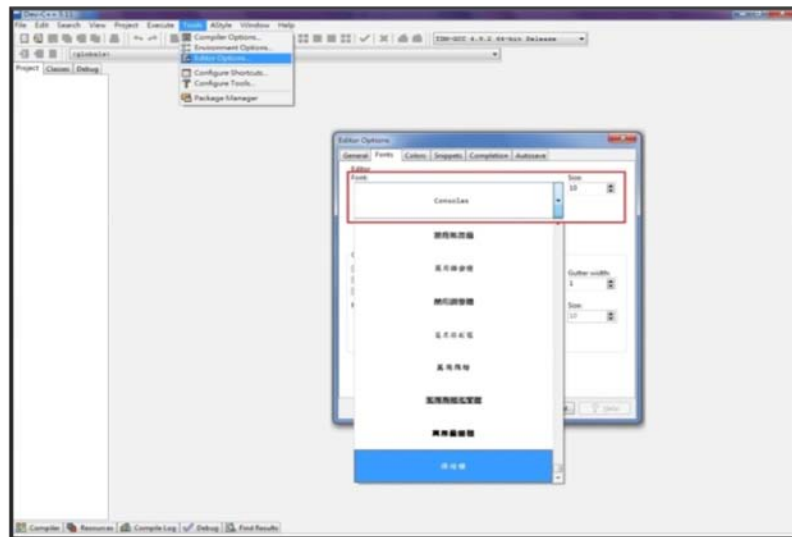


動作 2 - 初訂【格式字型】

## 🔗 參考附錄圖片(B2)



動作 3 - 重訂【選單語言】



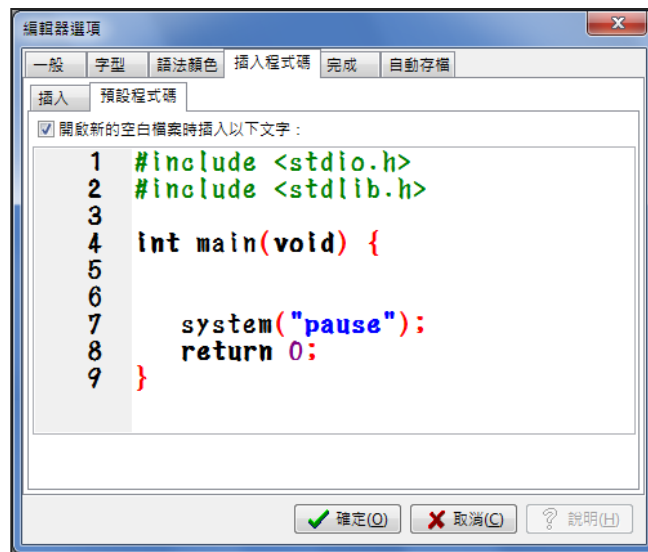
動作 4 - 重訂【格式字型】

### (3) 建立人工自訂範本・線上動作

點選功能表(工具・編輯器選項)彈出視窗，再執行(插入程式碼)下(預設程式碼)頁籤來進入編輯視窗。記得先行「勾選」開啟新的空白檔案時插入以下文字，最後進行「建立範本程式碼」作業。【參考右圖內容】

### (4) 建立原始碼，準備編寫程式

按下工具列(開新檔案・原始碼)圖示，產生「自訂範本・新文件」。點選功能表(檔案・另存新檔)，進行線上「文件命名」，最後進行「功撰寫能程式碼」。



【基礎架構】採用「一加一式」檔案管理

・ 原始碼 ( `.c` / `.cpp` ) --- Edit

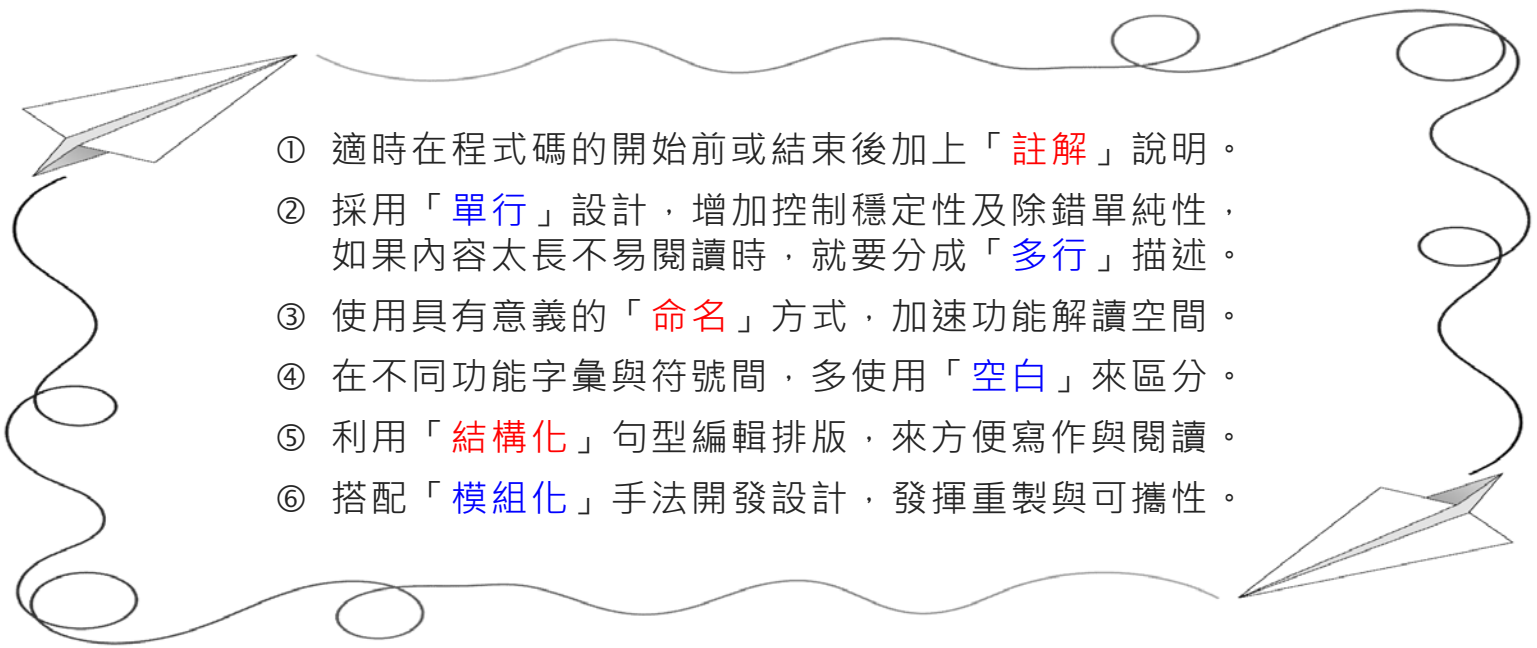
・ 執行檔 ( `.exe` ) --- Run

# \*研討節次.

## 1-3. 設計新手學前觀察重點

話說「**C語言**」採用「**線上函數集**·**Functions**」架構進行相關應用程式開發。此類技術特色在於將「**人類行為**」與「**電腦控制**」整合表述作業，此法看似設計動向單純，對初學應用者卻具語法解構困難度，只有完整累積「**函數文化**」使用經驗，才可徹底發揮「**線上工具**」最佳作業能力！

巨匠小幫手特別整理出下列多項重點法則給新手學員參考，培養專業化的技巧觀感，相信必能加速達成未來「**正規化設計**」學習任務。

- 
- ① 適時在程式碼的開始前或結束後加上「**註解**」說明。
  - ② 採用「**單行**」設計，增加控制穩定性及除錯單純性，如果內容太長不易閱讀時，就要分成「**多行**」描述。
  - ③ 使用具有意義的「**命名**」方式，加速功能解讀空間。
  - ④ 在不同功能字彙與符號間，多使用「**空白**」來區分。
  - ⑤ 利用「**結構化**」句型編輯排版，來方便寫作與閱讀。
  - ⑥ 搭配「**模組化**」手法開發設計，發揮重製與可攜性。

# \*研討節次.

## 1-4. 線上註解新增與移除化

程式「**註解** · **Comments**」功能表述，一般常用於「**加註功能說明**」及「**保留程式執行**」兩種時機。表述語法有「**區段**」與「**單行**」兩種變化指示，前者適用於「**多行完整**」狀態，使用彈性高，但不可進行巢狀架設應用，否則會發生線上參照失誤，進而出現編譯問題，而後者用於「**單行重點**」狀態，使用便利化，更有「**功能表指令**、**快速鍵**」線上支援，將可加速作業完成！

---

**區段註解** | 以 **/\*** 開頭，用 **\*/** 結尾

---

**單行註解** | 直接在起始處 **//**

---

\* 不論是上述何種「**符號字元**」，都要「**連續輸入**」完成，  
但可適時搭配「**自訂字元**」增加「**個人化風格**」表述效果！

Q · 如何使用「**單行**」註解，進行「**多行**」同步表述作業？

A · 點選 **功能表指令**（**編輯** · **設為註解 / 取消註解**）或按下 **快速鍵**（**Ctrl + /**）切換功能。



# \*研討節次.

## 1-5. 引用標題化【 #include 】

本次「[基礎課程](#) · [C語言](#)」採用「[重點參考](#)」語法檢查，發現設計使用了非內建工具，電腦就會要求引用「[標題檔](#) · [Header File](#)」技術來源，同時在程式編譯作業下，開發工具會將「[警告](#) · [Warning](#)」提示於「[編譯 / 除錯視窗](#)」中。考量「[進階課程](#) · [C++語言](#)」採用「[完整對照](#)」語法檢查，請提早配合引用動作。巨匠小幫手依據「[標題](#)、[函數](#)」設計關聯性整理於下區對照，提供各位學員線上參考！

---

<a href="#">stdio.h</a>		<a href="#">gets()</a> . <a href="#">puts()</a> . <a href="#">scanf()</a> . <a href="#">printf()</a> . <a href="#">getchar()</a> . <a href="#">putchar()</a> ...
<a href="#">stdlib.h</a>		<a href="#">system()</a> . <a href="#">srand()</a> . <a href="#">rand()</a> . <a href="#">malloc()</a> . <a href="#">free()</a> . <a href="#">abs()</a> . <a href="#">exit()</a> ...
<a href="#">conio.h</a>		<a href="#">getche()</a> . <a href="#">getch()</a> . <a href="#">putch()</a> ...
<a href="#">ctype.h</a>		<a href="#">toupper()</a> . <a href="#">tolower()</a> . <a href="#">isalpha()</a> . <a href="#">isupper()</a> . <a href="#">islower()</a> . <a href="#">isdigit()</a> . <a href="#">isspace()</a> ...
<a href="#">string.h</a>		<a href="#">strlen()</a> . <a href="#">strcpy()</a> . <a href="#">strcat()</a> . <a href="#">strrev()</a> . <a href="#">strupr()</a> . <a href="#">strlwr()</a> . <a href="#">strcmp()</a> ...
<a href="#">math.h</a>		<a href="#">pow()</a> . <a href="#">sqrt()</a> . <a href="#">fabs()</a> . <a href="#">fmod()</a> . <a href="#">ceil()</a> . <a href="#">floor()</a> . <a href="#">round()</a> ...
<a href="#">time.h</a>		<a href="#">time()</a> . <a href="#">ctime()</a> . <a href="#">localtime()</a> . <a href="#">gmtime()</a> . <a href="#">difftime()</a> ...

---

作業指令「[前置控制字元](#) · <#>」為「[前端機制](#)」程序字元。除了「[引用標題](#) · [#include](#)」用法外，線上還提供「[巨集定義](#) · [#define](#)」及「[編譯條件](#) · [#if](#)」特訂類型應用，這兩種「[專業開發](#)」控制句型將會在「[進階課程](#) · [C++語言](#)」中，搭配「[技術主題](#)」深入研討！

# \*研討節次.

## 1-6. 執行主函數【 main 】

每個程式語言都有一個「**起始執行區**」，做為進入程式之後的首要動作目標。

「**C語言**、**C++語言**」採用「**原始主函數**·**main**」為預訂命名。要增加執行完整性，就會在內部架設「資料型式、目標宣告、處理動作、控制句型」等線上互動化設計表述，最後搭配「線上註解」進行相對功能說明與碼值應用。

「**C語言**」採用「**原值型式**·**void**」進行精簡宣告，「**C++語言**」採用「**整數型式**·**int**」進行完整宣告，建議搭配「回傳指令值·**return 0**」來讓句型具備互動化表述，在完成回傳值動作時，會先將主函數相關記憶體應用空間釋放，接著結束本程式線上執行作業。

🔗 **C語言** · 主函數

```
void main( 外掛參數 ) {
```

線上互動化設計

```
}
```

🔗 **C++語言** · 主函數

```
int main( 外掛參數 ) {
```

線上互動化設計

```
return 0 ;
```

```
}
```

再提醒兩件事，首先是主函數中加註「**外掛參數**」，則表示將取用「**命令提示列**」模式後的線上「**目標資源**」，搭配內部程式碼進行功能作業。當發現程式行數過多且內容表述複雜時，設計師也可以線上開發「**自訂函數**」，將程式碼值進行功能「**整合管理**」，加速線上「**更新維護**」作業！

當設計者選用「**專案機制**」建立應用程式類型時，開發工具將會採用「**完整制式**」法則來建構範本程式碼值，相關內容參考如下...

#### 🔗 C語言 · 主函數

```
int main( int argc , char *argv[] ) {  
    線上互動化設計  
    return 0 ;  
}
```

#### 🔗 C++語言 · 主函數

```
int main( int argc , char** argv ) {  
    線上互動化設計  
    return 0 ;  
}
```

# \*研討節次.

## 1-7. 格式化存取【 scanf . printf 】

本節將進行「**格式化工具**」進行資料存取介紹！先從「**格式化**·**Format**」機制應用文化開始吧！所謂格式化即是電腦內建區域申請配置，依據「**資料型別**」及「**人工格式**」兩大關係指示的訂作空間存取法。格式區域分成字元(**%c**)、整數(**%d**)、浮點數(**%f**)及字串(**%s**)基底配置類型，對照「**資料型別**·**Types**」來進行關聯表述。型別資訊將在第二章節詳細說明，該類機制下「**工具函數**」作業語法，參考下區說明...

④ 資料輸入. **scanf** ( "格式資訊", 取址目標 )

- 註. (1) 相關系統內建值型式皆要加註「**取址字元**·**&**」，字串可以省略！  
(2) 只可進行「**區域空間**」表述，不負責「**資料顯示**」作業！  
(3) 可「**同步接收**」多重目標，請注意「**分隔**、**間距**」字元衝突問題！

④ 資料輸出. **printf** ( "格式資訊", 資料值目標 )

- 註. (1) 格式細分成「**資料文字**、**格式文字**、**特訂字元**」等三種設計關係解讀。  
(2) 採取「**本行輸出**」處理，加註「**特訂字元**」控制「**跳行**·**\n**」「**跳區**·**\t**」。  
(3) 格式設定「**區域空間**」外，亦可加註「**訂作參數字元**」，改造輸出結果！  
(4) 所謂資料值目標，可用「**資料原值**、**控制目標**、**參考運算式**」選擇架設！  
(5) 進行「**同步輸出**」多重目標時，請注意「**分隔**、**間距**」字元衝突問題！



相關格式化「**機制字元**」應用・課堂中配合「**實作範例**」題目・進行線上設計與功能指導說明。

⌚ 區域空間.	單一字元 <b>%c</b>	相對整數 <b>%d</b> . <b>%lld</b>	無號整數 <b>%u</b>	
	字元陣列 <b>%s</b>	8 進位值 <b>%o</b>	16進位值 <b>%x</b>	
	靜浮點數 <b>%f</b> . <b>%lf</b>	動浮點數 <b>%g</b>	指數處理 <b>%e</b>	

⌚ 格式字元.	文字對齊	靠左對齊 <b>-</b> . 靠右對齊 <b>+</b> ( <b>內訂</b> )	
	參考數據	空間位數 <b>M</b> / <b>N</b> / <b>.n</b> ( ps. <b>M</b> = <b>N</b> + <b>.n</b> )	
	特訂應用	完整進位 <b>#</b>   加註百分比 <b>%%</b> 填滿字元 <b>0</b> . <b>Space</b> ( <b>空白</b> )	

⌚ 特訂字元.	警告 <b>\a</b>	退位 <b>\b</b>	跳列 <b>\n</b>	
	跳頁 <b>\f</b>	跳區 <b>\t</b>	歸位 <b>\r</b>	
	終止 <b>\0</b> ( <b>數字</b> )	8 進位 <b>\o</b>	16進位 <b>\x</b>	
	特訂標符 <b>\</b> 關鍵標符 ( <b>單雙引號</b> 、 <b>問號</b> 、 <b>控制字元 ...</b> )			

# \*研討節次.

1-8. 程式設計、編譯、執行

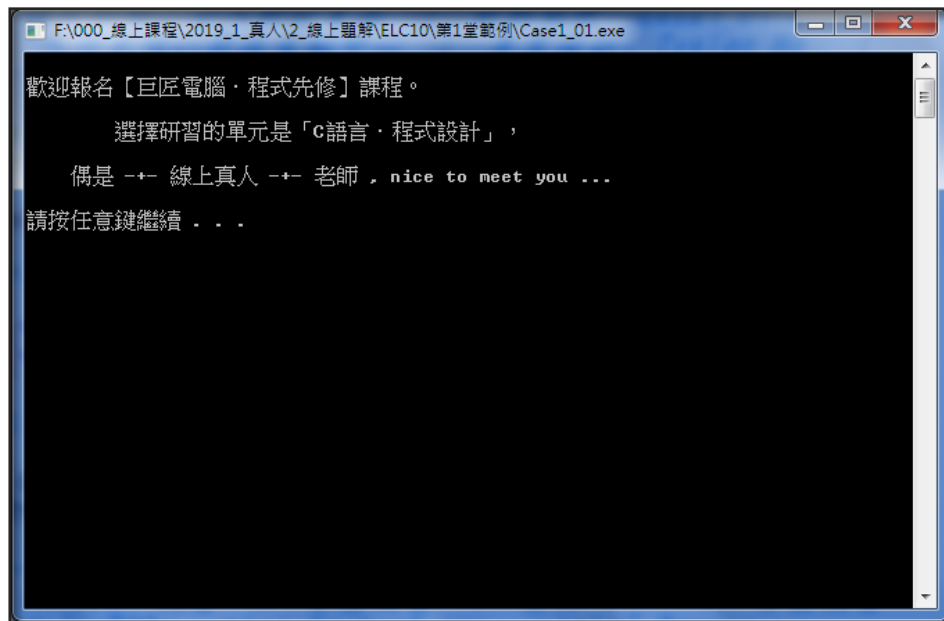
經過前頁各節次相關功能介紹後，接著把「**相關學習**」應用在「**範例題目**」上吧！先將應用程式採用「**原始碼**」作業類型建立「**新文件**」，同時體驗「**開發工具操作**」及探討「**線上設計語法**」。

① 建立原始碼並存檔 **Case1\_01.c** 後，輸入畫面程式碼，「**註解**」可保留輸入，加速設計動作。



```
Case1_01.c
1  // -----
2  //  |  程式功能. 登入歡迎畫面  |  範例編號. Case1_01.c  |
3  //  |  應用單位. 巨匠電腦中心  |  專案設計. Tsaigo Ho  |
4  // -----
5
6  #include <stdio.h>
7  #include <stdlib.h>
8
9  int main(void) {
10
11     // 登入歡迎畫面. 顯示訊息設計
12
13     puts("\n歡迎報名【巨匠電腦·程式先修】課程.\n");
14     puts("\t選擇研習的單元是「C語言·程式設計」,");
15
16     // 依照學名報名方式, 進行上課採用何種教學模式
17
18     char Online[]="偶是 -+- 線上真八 -+- 老師"; // 線上真八
19     printf("\n%30s , nice to meet you ...\n\n",Online);
20
21     /* -----
22     char Room[]="偶是 -+- 實體教室 -+- 老師"; // 實體教室
23     printf("\n%30s , nice to meet you ...\n\n",Room);
24     ----- */
25
26     system("pause");
27     return 0;
28 }
```

- ② 可選用( 功能表 · 執行 · 編譯 )或按下( 工具列 · 編譯 )或( 快速鍵 · F9 )任一作法，進行「語法檢查」及追蹤「工具連結」技術來源，線上建製執行檔( .exe )。
- ③ 可選用( 功能表 · 執行 · 執行 )或按下( 工具列 · 執行 )或( 快速鍵 · F10 )任一作法，進行「應用程式」線上「設計功能」測試，同步編輯相關功能原始碼( .c / .cpp )。



-----【線上技術 · 提醒時間】-----

可選用( 功能表 · 執行 · 編譯並執行 )  
或按下( 工具列 · 編譯並執行 )  
或( 快速鍵 · F11 )任一作法，  
將步驟②③同步作業，加速查閱結果。



# \*研討節次.

1-9. 範例題目・實作研討

| 實作研討題目 | 印出「書籤版面」，利用「格式化工具」進行多元化設計。

提示・將第五行「留言者」內容，採取訂作「目標格式」輸出。

【範例編號：Case1\_02.c】



懂得分享，人生是彩色的！  
生命的能量活水，  
從成長情緒中湧出！  
  
逍遙學長。材哥題  
請按任意鍵繼續 . . .

| 實作研討題目 | 輸入出生「民國年份」後，求取對應「西元年份」及「現在年齡」。

提示・年齡採取「虛歲」計算方式，不參考出生月份及日期。

出生民國年份：69  
對應西元年份：1980，今年 39 歲  
請按任意鍵繼續 . . .



【範例編號：Case1\_03.c】

| 實作研討題目 | 同步登錄「國文、英文、數學」學測成績，進行三科「總分、平均」計算報告。

提示・將「總分、平均」求取項目結果值後，再進行「指定位數」格式輸出。

【範例編號：Case1\_04.c】



----- 登錄學測成績 -----

科目別：國文・英文・數學

92 85 73

----- 計算學測分數 -----

三科總分：250

三科平均： 83.3

請按任意鍵繼續 . . .



# 問卷

<http://www.pcschoolonline.com.tw>

開課查詢

免費體驗專區

課程總覽

專業師

1

學員專區

講師專區



➤ 課程檔案下載：

學員的「上課教材」，下載檔案為壓縮檔 ([解壓縮操作步驟](#))。  
如無法觀看上課教材，請安裝 [PDF閱讀軟體](#)。

公告專區

我的課表

課程劃位

取消劃位

2

課程檔案下載

自107年1月1日起，課程錄影檔由180天改為365天(含)內無限次觀看 (上課隔日18:00起)。

問  
卷

上課日期	課程名稱	課程節次	教材下載		
2017/12/27 2000 ~ 2200	線上真人-ZBrush 3D動畫造型設計	18	<a href="#">上課教材</a>	<a href="#">錄影檔</a>	<a href="#">課堂問卷</a>
2017/12/20 2000 ~ 2200	線上真人-ZBrush 3D動畫造型設計	17	<a href="#">上課教材</a>	<a href="#">錄影檔</a>	
2017/12/18 2000 ~ 2200	線上真人-ZBrush 3D動畫造型設計	16	<a href="#">上課教材</a>	<a href="#">錄影檔</a>	



巨匠線上真人

[www.pcschoolonline.com.tw](http://www.pcschoolonline.com.tw)