

# 第一週

王子銓, 陳毅軒, 吳尚龍

電機通訊程式設計

February 19, 2024

# Outline

- 1 上課時間、評分、規則
- 2 Domjudge 使用，以及作業繳交
- 3 課程內容

## 上課時間

- 上課時間: 每週一晚上 6.~ 9.
- 期中考: April 15
- 期末考: June 17
- 點名: 第二週開始後, 每次上課 6.20 點名

缺席次數與正課一併計算, 若請假請上 Ecourse2 申請, 需有正當理由

# 評分標準

## 評分標準

### 與正課一併計分

- 作業: 共 30%
- 筆試: 期中 20%, 期末 20%
- 上機: 期中 20%, 期末 20%
- 加分: 作業若於上課日晚上 9: 00 前全數通過 Domjudge 評測且找助教 Demo 完畢, 總成績 + 0.5

上限 100 分


## 課堂規則

- 每周均有作業，需在下周晚上 6:00 前上傳至 Ecourse2 上
- 作業檔案命名正確
  - 程式碼檔案名稱需對應 DomJudge 上的題目名稱
    - 格式為 [學號]\_[題目編號].c
    - Ex: 410430049\_A.c
  - 需將所有程式碼壓縮為.zip 檔
    - 格式為 [學號]\_W[當前周數].zip
    - W 需大寫、沒有空格，檔案名稱須完全符合格式
    - Ex: 410430049\_W1.zip
- 作業請勿抄襲！抄襲者與被抄襲者，當週作業成績以 0 分計算


- 作業網站網址: [pgds.csie.io](https://pgds.csie.io)  
~~需從學校內的網域才可使用，連學校的網路可直接使用，~~  
~~若要使用外部網路，請使用 VPN，VPN 教學在 Ecourse2 公佈欄。~~  
經過助教的努力，現在不用掛 VPN 了
- Domjudge 帳號密碼皆為學號，本學期所有作業以及上機考皆使用此網站
- Domjudge 使用教學可參考 [Domjudge 使用教學](#)
- 本堂課程僅可使用 C 語言


# 作業繳交範例


壓縮完上傳的檔案要長這樣


 410430049\_W1.zip

將其解壓縮後會出現相同名稱的資料夾，資料夾裡應需要有每題的.c 檔

 410430049\_A.c

 410430049\_B.c

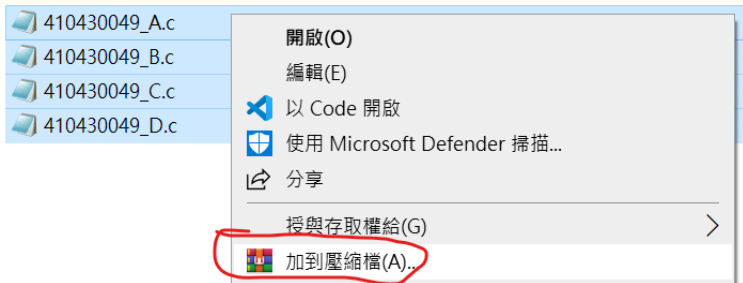
 410430049\_C.c

 410430049\_D.c

# 檔案壓縮教學

使用 **window/mac**:

將全部檔案全選後，按滑鼠右鍵，選擇**加到壓縮檔**





確認一下檔案名稱，之後選擇.zip 以及壓縮即可



- 作業格式錯誤者，當週作業以 0 分計算
- 需在 Domjudge 通過題目，且繳交到 Ecourse2 的程式碼皆需可通過，才有分數
- 沒有通過的題目不需上傳至 Ecourse2

# 註解

用來說明檔名、變數作用、function 功能。

註解有兩種

- ① `//` 只能單行註解
- ② `/*.....*/` 可以多行註解

```
1 // 語法
2 printf("hello World");
3 /*
4 輸出
5 Hello World
6 This is comment
7 */
```

推薦使用 vscode 可以 `ctrl + /` 多行註解

用於引入函式庫 (Library) 的標頭檔，標頭檔內有別人寫好的函式庫，可以直接拿來用。

例如: `stdio.h` (standard input/output header) 掌管標準輸入輸出。

```
1 // 語法
```

```
2 #include <stdio.h>
```

```
3 #include <stdlib.h>
```

```
4 #include <string.h>
```

```
5 #include <stdbool.h>
```

# 變數與資料型態

爲了存放數值，我們必須宣告變數，每種變數的記憶體大小不盡相同，且變數存在的記憶體空間都有一個地址。

C 語言的資料型態大致分爲以下幾種

- int (整數)
- float (浮點數)
- char (字元)
- array (陣列)
- pointer (指標)
- structure (結構)

```
1 // 語法
```

```
2 int x = 1; // 宣告變數名稱爲 x、型態爲 int，並且賦值 1 給它
```

# statement 敘述式

執行一個動作 (action)，以分號 (;) 作為結束符號 (statement terminator)。

```
1 // 語法
2 int a = 1;
3 int b = 48762;
4 printf("%d", a + b);
```

## scanf

- 使用 & 取得各個變數的記憶體位置，並作為參數傳入 scanf 內，這樣才知道要把數值存在哪裡。
- 格式化字串如果遇到 % 會進行讀入。
- % 後面可以用不同英文字代表要讀入的型態。
- 例如: %d 代表整數, %f 代表浮點數, %c 代表字元, %s 代表字串。

```
scanf ("格式化字串", &變數1, &變數2, ...);
```

## printf

```
1 printf ("格式化字串", 變數1, 變數2,...);
```

用法與 scanf 大同小異，因為只要輸出變數的值，所以不用 &，直接將變數的數值傳入 printf 即可。另外，如果是輸出浮點數，可以在 % 跟 f 中間加入 [數字]，來指定輸出小數點後第幾位。

```
1 // 語法  
2 float x = 4.8763;  
3 printf("%.2f\n", x); // 輸出 4.87
```



# 良好的 coding style

- 每個字節間以空白隔開
- 出現逗號後會多一個空格
- 程式碼每行不宜太長
- 多寫註解

```
1 // 多寫註解，說明自己的程式碼是在做什麼的
```

```
2 // 基本上用空白隔開每個字節，除了逗號前方以及結尾的分號
```

```
3 int a = 8, b = 9;
```

雖然不會影響程式的結果，但能注意還是盡量注意，良好的 coding style 會讓程式碼的可讀性上升

- '+', '-', '\*', '/' 與一般的數學式一樣，可以拿來直接用
- 當兩個不同型態的變數做運算時，結果會依照前面變數的型態

# main 主函式

每個程式都必須有的函式，並且同個 project 裡只能存在一個。

```
1 // 語法
2 int main(){
3     printf("Hello World");
4     return 0;
5 }
```

函式後面會有大括號，包住接下來要執行的 code。

# 區塊 block {}

每個函式後面必有的東西，但不限於只能用一個，也可以在區塊裡放更小的區塊。

```
1 // 例子
2 int main(){
3     int a = 1;
4     {
5         int a = 2;
6         printf("%d\n", a); // 輸出2
7     }
8     {
9         int a = 3;
10        // int a = 4; 會報錯
11        printf("%d\n", a); // 輸出3
12    }
13    printf("%d\n", a); // 輸出1
14    return 0;
15 }
```

同個區塊內只能宣告一次同個變數名稱，並且在運算時，如果遇到同名的變數名稱 (不同 block)，會以區域小的變數為主。

# if 判斷式

執行判斷 (judge)，接著在內寫入當判斷成立時，所要執行的內容。

```
1 // 語法
2 int a = 1;
3 if( a > 1 ){
4     printf("a bigger than 1");
5 }
```

# if else 判斷式

執行判斷 (judge)，接著在內寫入當判斷結果為 True 時，所要執行的內容；如果判斷結果為 false 時，執行 else 內的敘述式。

```
1 // 語法
2 int a = 1;
3 if( a > 1 ){
4     printf("a bigger than 1");
5 }
6 else {
7     printf("a less than or equal to 1");
8 }
```

## else if 判斷

當有連續判斷的內容，可以接續判斷。

```
1 // 語法
2 int a = 1, b = 1;
3 if( a > 1 ){
4     printf("a is bigger than 1");
5 }
6 else if (b > 1){
7     printf("a is less than or equal to 1");
8     printf("b is bigger than 1");
9 }
10 else{
11     printf("a and b are less than or equal to 1");
12 }
```

# 判斷式注意事項

當有連續判斷的內容，if, else if, else 中只會進入第一個符合的 block

```
1 // 語法
2 int a = 10;
3 if( a > 1 ){
4     printf("a is bigger than 1");
5 }
6 else if (a > 5){
7     printf("a is less than 5 and is bigger than 1");
8 }
9 else {
10     printf("a is no more than 1");
11 }
12 // a bigger than 1
```

雖然 a 也符合 else if 中的， $a > 5$  的條件，但因為已經進入前面 if 的那個 block 裡面，因此後面的條件變不會觸發