

105學年 資料結構

Class 1 : Coding Style

你會不會寫程式？

下面有幾個是你會的？

- 陣列(Array)
- 堆疊(Stack), 佇列(Queue)
- 指標(Pointer), 位址(Address)
- 鏈結串列(Linked List)
- 結構(Struct), 樣板(Template), 類別(Class)
- 樹狀結構(Tree Struct)

你說你不會

教你幾招

- 要會叫狗(Google)找資料。
- 用紙和筆畫個流程圖。
- 遇到Bug想辦法解決。
- 找同學、學長討論。
- 自己的程式自己寫!

Coding style

為何要了解Coding Style

你說：我自己寫的程式自己看得懂就好。

但是在這世上還是有一個人要看你寫的程式碼

你自己

****別相信明天的你會看得懂自己現在寫的Code****

Coding Style

- 變數命名

- 縮排

- 換行

- 註解

變數命名

👁 有意義的命名

```
2 string full_name;  
3 string first_name, last_name;  
4 int money;
```

```
2 string n;  
3 string f, l;  
4 int m;
```

單字過長就簡寫吧！ex: number => num

👁 命名方式：

蛇底式命名法(Snake Case)	snake_case
駝峰式命名法(Camel Case)	CamelCase camelCase

縮排

為何要縮排？

- Beautiful, 美觀很重要
- 容易檢查括號有無成對
- 程式碼區塊明瞭

```
1 #include <iostream>
2
3
4 int main()
5 {
6   for(int i=1;i<=10;i++)
7   {
8     cout<<i*2<<endl;
9   }
10 }
```

```
2 #include <iostream>
3
4 int main()
5 {
6     for(int i = 1; i <= 10;i++)
7     {
8         cout<<i*2<<endl;
9     }
10 }
```


換行

- 寫code就像寫文章，要段落分明。

```
4 if(check<10){check++;cout<<check;}
```

```
4 if (check < 10)
5 {
6     check++;
7     cout << check;
8 }
```



註解

- 註解可有可無，有註解就是方便閱讀
- 簡潔說明該段程式碼在做什麼事情即可
- 避免多餘的註解

```
14  if (i == 0) { // 如果i等於0
15      sum = 0; // 將sum設為0
16  }
```

```
2  int s[]; // 分數
3
4  int sum; // 總分
5
6  // 算總分
7  for (int i = 0; i < 5; ++i)
8      sum += s[i];
9  cout << sum/5 << endl;
```


總而言之



別怠惰拉

沒縮排沒換行助教們不看