

# C++ Programming HW #3 - Study Group Reservation

B11901123 張鈺源

## 1. 前言

靈感源自系上提供的讀書會。因為目前讀書會的預約系統皆使用 Google Form，所以我想使用課堂所學的 Object-oriented programming 編寫一套計算機程式課的預約系統，系統可供助教填寫每周方便的時段，並可做修正、存取；而學生也可上預約系統登記每周適合的時段。此程式有兩個 interface，一個是 TA interface，另一個是 Student interface，之間採用輸入身分與 password（後續詳細說明）隔開。程式大致規劃如下：

1. 使用者輸入身分（TA 或 Student）。
2. 以 password 判別 TA 身分；以 ID 與 Professor 判別 Student 身分。
3. TA 輸入每周所開的讀書會時段；Student 輸入欲參加的時段。
4. 資料可進行新增、儲存、修改。

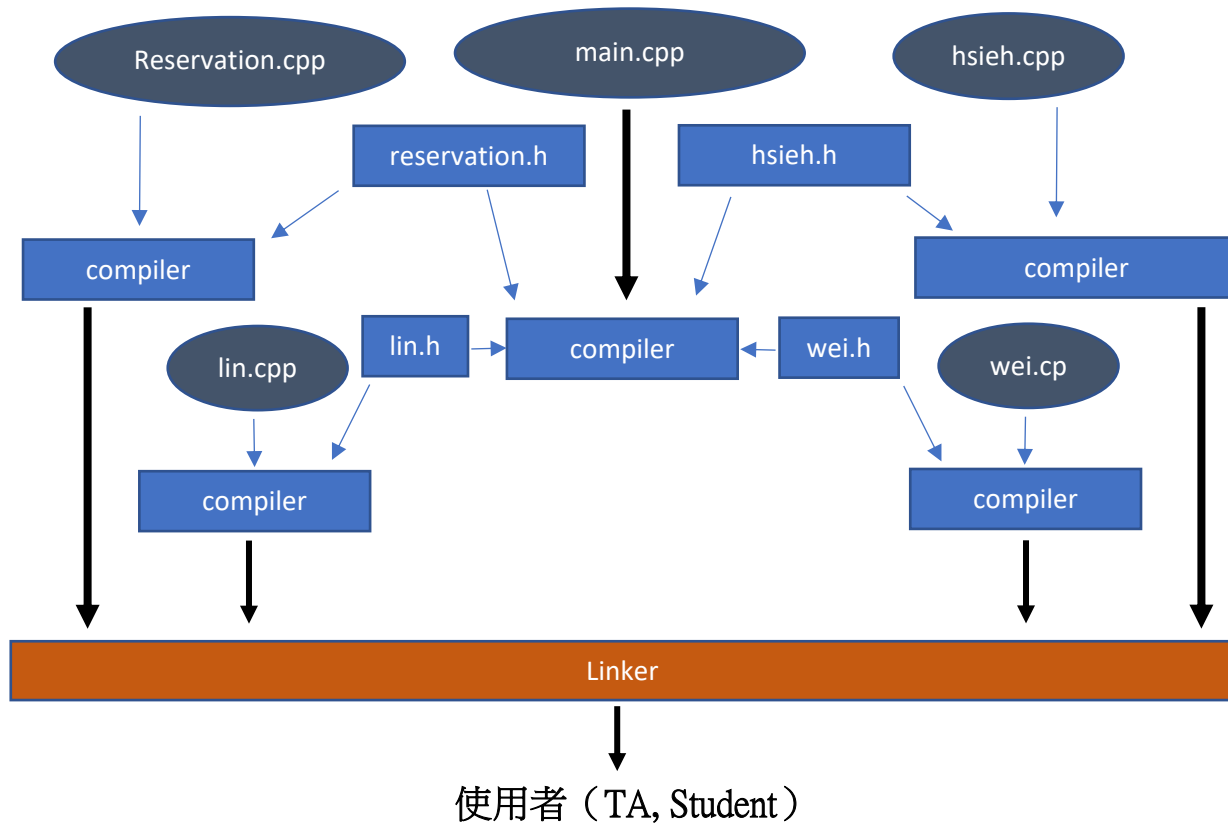
程式簡介：

1. 宣告的 class: reservation（Student ID、professor）、wei（魏宏宇教授的班級）、hsieh（謝宏昀教授的班級）、lin（林宗男教授的班級）。
2. 使用的 syntax: functions、if、else if、switch、do while、pass by reference、friend、constant data member。
3. 使用的 control statement: selection statement、nested control statement。

## 2. 程式解決了什麼問題

1. 一個程式能擁有兩個 interface → 供兩種身分獨立使用，但 private 資料相通。
2. 一班的 TA 無法 access 別班的 TA interface（需要輸入各班的 Password 方可進入）→ 防止亂新增或修改時段。
3. TA 輸入每周的讀書會時段後，還可以修改 → Flexibility。
4. 學生進入 student interface 前需輸入 ID 與 professor，若 ID 非 111 學年度電機系學生，或者 professor 非三位計程教授之一，不予以進入 → 防止非相關人士預約計程讀書會。

### 3. Class 架構與內容



#### 1. 各個 class 內容說明

Reservation	.h	<pre> class reservation {     public:         void setData(string, string);         void setID(string);         void setTeacher(string);         string getID();         string getTeacher();         void printID();         void printTeacher();          private:             string student_id;             string teacher; }; </pre> <p>基本函式: 資料設置、 讀取、列印</p>
	.cpp	<p>Ex. setID: 判斷 ID 是否為 B11901 開頭(電機系大一)。 setTeacher: 判斷教授是否為 wei, hsieh 或 lin。</p> <pre> void reservation::setID(string s1) {     student_id = ((s1[0]=='B'    s1[0]=='b') &amp;&amp;         s1[1]=='1' &amp;&amp; s1[2]=='1' &amp;&amp;         s1[3]=='9' &amp;&amp; s1[4]=='0' &amp;&amp; s1[5]=='1')? s1 : "0"; }  void reservation::setTeacher(string s2) {     teacher = (s2=="Wei"    s2=="wei"    s2=="Hsieh"    s2=="hsieh"            s2=="Lin"    s2=="lin")? s2 : "0"; } </pre>

Wei	.cpp	<ol style="list-style-type: none"> <li>1. 視 reservation 為 friend，故 reservation 可讀取 private 的時段。</li> <li>2. Pass 時段 by reference，若使用者新增或修改資料，最後顯示的時段會被 update。</li> </ol> <pre> class wei {     friend class reservation;     friend void w_display(); public:     wei(const string&amp;, const string&amp;, const string&amp;, const string&amp;, const string&amp;);     void w_print() const;  private:     const string w_mon_time;     const string w_tue_time;     const string w_wed_time;     const string w_thu_time;     const string w_fri_time; }; </pre>
	.h	<ol style="list-style-type: none"> <li>1. Constant member initializer</li> <li>2. Member function: Print 每一班的課表</li> </ol> <pre> wei::wei(const string &amp;mon1, const string &amp;tue1, const string &amp;wed1, const string &amp;thu1, const string &amp;fri1) : w_mon_time(mon1),   w_tue_time(tue1),   w_wed_time(wed1),   w_thu_time(thu1),   w_fri_time(fri1) { }  void wei::w_print() const {     cout &lt;&lt; setw(2) &lt;&lt; " This is the current schedule" &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; "-----" &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " MON: ";     cout &lt;&lt; w_mon_time &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " TUE: ";     cout &lt;&lt; w_tue_time &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " WED: ";     cout &lt;&lt; w_wed_time &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " Thu: ";     cout &lt;&lt; w_thu_time &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " Fri: ";     cout &lt;&lt; w_fri_time &lt;&lt; endl &lt;&lt; endl; } </pre>
Hsieh、Lin	.cpp	<p>與 Wei Class 相似。</p> <pre> hsieh::hsieh(const string &amp;mon1, const string &amp;tue1, const string &amp;wed1, const string &amp;thu1, const string &amp;fri1) : h_mon_time(mon1),   h_tue_time(tue1),   h_wed_time(wed1),   h_thu_time(thu1),   h_fri_time(fri1) { }  lin::lin(const string &amp;mon1, const string &amp;tue1, const string &amp;wed1, const string &amp;thu1, const string &amp;fri1) : l_mon_time(mon1),   l_tue_time(tue1),   l_wed_time(wed1),   l_thu_time(thu1),   l_fri_time(fri1) { } </pre>
	.h	<p>與 Wei Class 相似。</p> <pre> void hsieh::h_print() const {     cout &lt;&lt; setw(2) &lt;&lt; " This is the current schedule" &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; "-----" &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " MON: ";     cout &lt;&lt; h_mon_time &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " TUE: ";     cout &lt;&lt; h_tue_time &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " WED: ";     cout &lt;&lt; h_wed_time &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " Thu: ";     cout &lt;&lt; h_thu_time &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " Fri: ";     cout &lt;&lt; h_fri_time &lt;&lt; endl &lt;&lt; endl; }  void lin::l_print() const {     cout &lt;&lt; setw(2) &lt;&lt; " This is the current schedule" &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; "-----" &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " MON: ";     cout &lt;&lt; l_mon_time &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " TUE: ";     cout &lt;&lt; l_tue_time &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " WED: ";     cout &lt;&lt; l_wed_time &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " Thu: ";     cout &lt;&lt; l_thu_time &lt;&lt; endl;     cout &lt;&lt; setw(2) &lt;&lt; " Fri: ";     cout &lt;&lt; l_fri_time &lt;&lt; endl &lt;&lt; endl; } </pre>

## 4. Main.cpp 內容介紹

```
string ID;
string Teacher;
string w_mon1, w_tue1, w_wed1, w_thu1, w_fri1;
string h_mon1, h_tue1, h_wed1, h_thu1, h_fri1;
string l_mon1, l_tue1, l_wed1, l_thu1, l_fri1;

cout << "****Welcome to Computer Programming Study Group Reservation Page****";
_getch();
cout << endl << endl;
int TorS; //choose between TA or student
```

```
#include <iostream>
#include <string>
#include <iomanip>
#include <conio.h>
#include "reservation.h"
#include "wei.h"
#include "hsieh.h"
#include "lin.h"
using namespace std;
```

1. 宣告 string: ID、Teacher、各班的時程（星期一～五）。
2. 宣告 integer: TorS（TA 或 Student）。
3. 使用 `_getch()`：pause 程式。
4. Include libraries 以及 .h 檔。

### 1) TA Interface

```
do
{
    cout << "Please select: " << endl;
    cout << "1. TA" << endl;
    cout << "2. Student" << endl << endl;
    cout << "Press -1 to cancel" << endl << endl;

    cin >> TorS;

    system("CLS");
}
```

```
if(TorS==1)
{
    cout << "See you next time." << endl;
    return 0;
}

cout << "****TA Interface****";
cout << endl << endl;
```

1. 選擇身分。
2. 若輸入「-1」，程式結束。
3. 使用 `system("CLS")`：clear screen。

```
/*TA Interface*/
if(TorS==1)
{
    cout << "Which professor?" << endl;
    cout << "1. Wei" << endl;
    cout << "2. Hsieh" << endl;
    cout << "3. Lin" << endl << endl;

    int prof;
    cin >> prof;
    cout << endl;

    switch (prof)
    {
        case 1: /*Wei*/
        {
            cout << "Enter TA password" << endl;
            string pw;
            cin >> pw; //password is hywei
            int reedit;

            default:
            {
                cout << endl << "Invalid Professor, press any key to exit.";
                _getch();
                system("CLS");
            }
            else
            {
                cout << endl << "Incorrect password, please try again." << endl;
                _getch();
                system("CLS");
            }
        }
    }
}
```

1. 若身分為 TA，進一步輸入是哪一位教授的 TA，並輸入密碼。（密碼為教授英文名字的 initials + surname）
2. 若查無教授，離開 TA interface，回到主頁面。
3. 若 password 正確，可新增或修改資料；若密碼錯誤，回到主頁面。

```

if(pw=="hywei")
{
    do
    {
        system ("CLS");
        cout << "****TA Interface****" << endl << endl;
        wei schedule_w(w_mon1, w_tue1, w_wed1, w_thu1, w_fri1);
        schedule_w.w_print();

        cout << "Enter:" << endl;
        cout << "1. Edit" << endl;
        cout << "2. Quit" << endl << endl;
        int temp;
        cin >> temp;

        system ("CLS");
        cout << "****TA Interface****" << endl;

        if(temp==2)
            continue;

        else if(temp==1)
        {
            cout << endl;
            instructions();

            cout << "Mon: ";
            cin >> w_mon1;

            cout << "Tue: ";
            cin >> w_tue1;

            cout << "Wed: ";
            cin >> w_wed1;

            cout << "Thu: ";
            cin >> w_thu1;

            cout << "Fri: ";
            cin >> w_fri1;
            cout << endl;
        }
    }
}

```

1. 以 case 1（魏班）為例，若 password 輸入正確，顯示目前的 schedule，並且請使用者選擇要編輯或離開。
2. 若輸入「2」，離開 TA interface，回到主頁面。
3. 若輸入「1」，分別輸入每周讀書會的時間。

```

system ("CLS");
cout << "****TA Interface****" << endl << endl;
wei schedule_w(w_mon1, w_tue1, w_wed1, w_thu1, w_fri1);
schedule_w.w_print();

cout << "Enter:" << endl;
cout << "1. Save" << endl;
cout << "2. Edit" << endl << endl;
cin >> reedit;
cout << endl;

}
}while(reedit==2);

if(reedit==1)
{
    system ("CLS");
    cout << "****Computer Programming Study Group Reservation Page****" << endl << endl;
    break;
}
}

```

1. Print 目前的 schedule。
2. 若輸入「2」，再修改 schedule。
3. 若輸入「1」，資料儲存，離開 TA interface，回到主頁面。

## 2) Student Interface

```

reservation No1;
system ("CLS");
cout << "****Student Interface****" << endl << endl;
cout << "Please enter your ID: ";
cin >> ID;
No1.setID(ID);

cout << "Please enter your professor: ";
cin >> Teacher;
No1.setTeacher(Teacher);

if(No1.getID()=="0" || No1.getTeacher()=="0")
{
    cout << endl << "Invalid ID or Teacher, please try again.";
    _getch();
    system ("CLS");
}

```

1. Instantiate object No1 of class reservation。
2. 輸入 ID、Teacher。
3. 若 ID 或 Teacher invalid，回到主頁面。

```

cout << "Would you like to join this week's study group?" << endl;
cout << "Enter:" << endl;
cout << "1. Yes" << endl;
cout << "2. No" << endl << endl;
int YorN;
cin >> YorN;

```

不參加讀書會

參加讀書會

```

if(YorN==2)
{
    cout << endl << "Press any key to exit.";
    _getch();
    system ("CLS");
}

```

```

else if(YorN==1)
{
    cout << "Please pick a time:" << endl;
    cout << "|NOTE: NA means \"Not Available\"|" << endl << endl;
    cout << "Enter:" << endl;
    cout << "1. Monday" << endl;
    cout << "2. Tuesday" << endl;
    cout << "3. Wednesday" << endl;
    cout << "4. Thursday" << endl;
    cout << "5. Friday" << endl << endl;
    int pick_a_time;
    cin >> pick_a_time;
}

```

1. 若輸入「2」，不參加讀書會，離開 Student interface，回到主頁面。
2. 若輸入「1」，螢幕顯示目前的 schedule，使用者任選一天。(1~5：星期一~5)

```

switch(pick_a_time)
{
    case 1:
    {
        system ("CLS");
        cout << "****Student Interface****" << endl << endl;
        cout << setw(2) << "-----" << endl;
        cout << setw(2) << "          *Confirmation data*" << endl << endl;
        cout << "    ID:";
        N01.printID();
        cout << endl;
        cout << "    Prof:";
        N01.printTeacher();
        cout << endl;
        cout << "    Time:    " << "Monday " << w_mon1 << endl;
        cout << setw(2) << "-----" << endl << endl;

        instructions2();
        break;
    }
}

```

1. 以魏班星期一為例，最後 print confirmation data，內容包含學生 ID、professor、時間。

## 5. 如何執行程式

### 1) TA

```

****Welcome to Computer Programming Study Group Reservation Page****

Please select:
1. TA
2. Student

Press -1 to cancel

1

```

1

```

****TA Interface****

Which professor?
1. Wei
2. Hsieh
3. Lin

1

Enter TA password
hywei

```

<Note> Passwords

1. Wei: hywei
2. Hsieh: hyhsieh
3. Lin: tnlin

2

```

****TA Interface****

This is the current schedule
-----
MON:
TUE:
WED:
Thu:
Fri:

Enter:
1. Edit
2. Quit

1

```

← 一開始仍空白

3

```

****TA Interface****

Insert available hours
eg. 6PM-7PM
|NOTE: insert NA for "Not Available"|

Mon: 7PM-9PM
Tue: NA
Wed: 1PM-2PM
Thu: NA
Fri: NA

```

← 自行輸入

4

```

****TA Interface****

This is the current schedule
-----
MON: 7PM-9PM
TUE: NA
WED: 1PM-2PM
Thu: NA
Fri: NA

Enter:
1. Save
2. Edit

1

```

Next: 回到主頁面 →

<Note> 若輸入「2」，回到步驟 2

## 2) Student

```

****Computer Programming Study Group Reservation Page****

Please select:
1. TA
2. Student

Press -1 to cancel

2

```

1

\*\*\*\*Student Interface\*\*\*\*

Please enter your ID: B11901123  
Please enter your professor: Wei

自行輸入

2

\*\*\*\*Student Interface\*\*\*\*

This is the current schedule

-----  
MON: 7PM-9PM  
TUE: NA  
WED: 1PM-2PM  
Thu: NA  
Fri: NA

Would you like to join this week's study group?

Enter:

1. Yes  
2. No

1

<Note> 若輸入「2」，回到主頁面

3

\*\*\*\*Student Interface\*\*\*\*

This is the current schedule

-----  
MON: 7PM-9PM  
TUE: NA  
WED: 1PM-2PM  
Thu: NA  
Fri: NA

Please pick a time:

|NOTE: NA means "Not Available"|

Enter:

1. Monday  
2. Tuesday  
3. Wednesday  
4. Thursday  
5. Friday

3

4

\*\*\*\*Student Interface\*\*\*\*

-----  
\*Confirmation data\*

ID: B11901123  
Prof: Wei  
Time: Wednesday 1PM-2PM  
-----

Thank you for reserving, press any key to exit.

Exit: 回到主頁面



!

```
****Student Interface****
```

```
Please enter your ID: C11901123
```

```
Please enter your professor: Iew
```

```
Invalid ID or Teacher, please try again.
```

若輸入非「111 學年電機大一學號」或非「111 學年電機系計程教授」

→ 回到主頁面。

## 4. 結語

最後一份計程個人作業終於完結了！本程式使用 OOP 編寫，使各個 class 的 data member 和 member function 可重複使用，且可以選擇性地隱藏一些資料。Reservation 的 class 存取學生的 ID 與教授（皆為 private），而 wei、hsieh 和 lin class 個別存取每周讀書會的 schedule（也為 private）。寫完之後，我領略到編寫較大的程式時，使用 class 的好處：將 header、implementation 與 main 分別放在不同 file，程式較為井然有序，否則光是 debug 就非常頭痛了。

目前的預約系統只支援 TA 新增或修改 schedule，而一個時段可以無上限被選擇。這顯然不太符合現實狀況。因此，未來我想加入 maximum reservation limit，當某一個時段超過 5 人預約時，便會額滿。若之後還有人想預約，schedule 的那一時段便會顯示「Full」。再者，我也想將擴增預約系統，使其也可支援其他課程的讀書會，一個預約系統，可以預約多個課程！最後，我想將其美化、優化，並轉換成 html 網頁版，因此所有電機系的學生皆可使用專屬的讀書會預約網站！我目前想到的困難點是如何儲存資料縱使關閉網頁，因為不可能每次都要請 TA 重新輸入，完全沒有效率。

以上是我的 Study Group Reservation，謝謝。