多元選修－翻轉機器人

10917 王翊鑫

目錄

摘要

- 這份報告在敘述我在「翻轉機器人」的課程中學習程式語言Ｃ＋＋的開始、過程、成果與省思。過程中，老師先教導概念，接著讓我們去「高中生程式解題系統」做題目，使我們不要只是聽完概念似懂非懂或是直接看別人打好的程式變成背程式，而是學會運用。

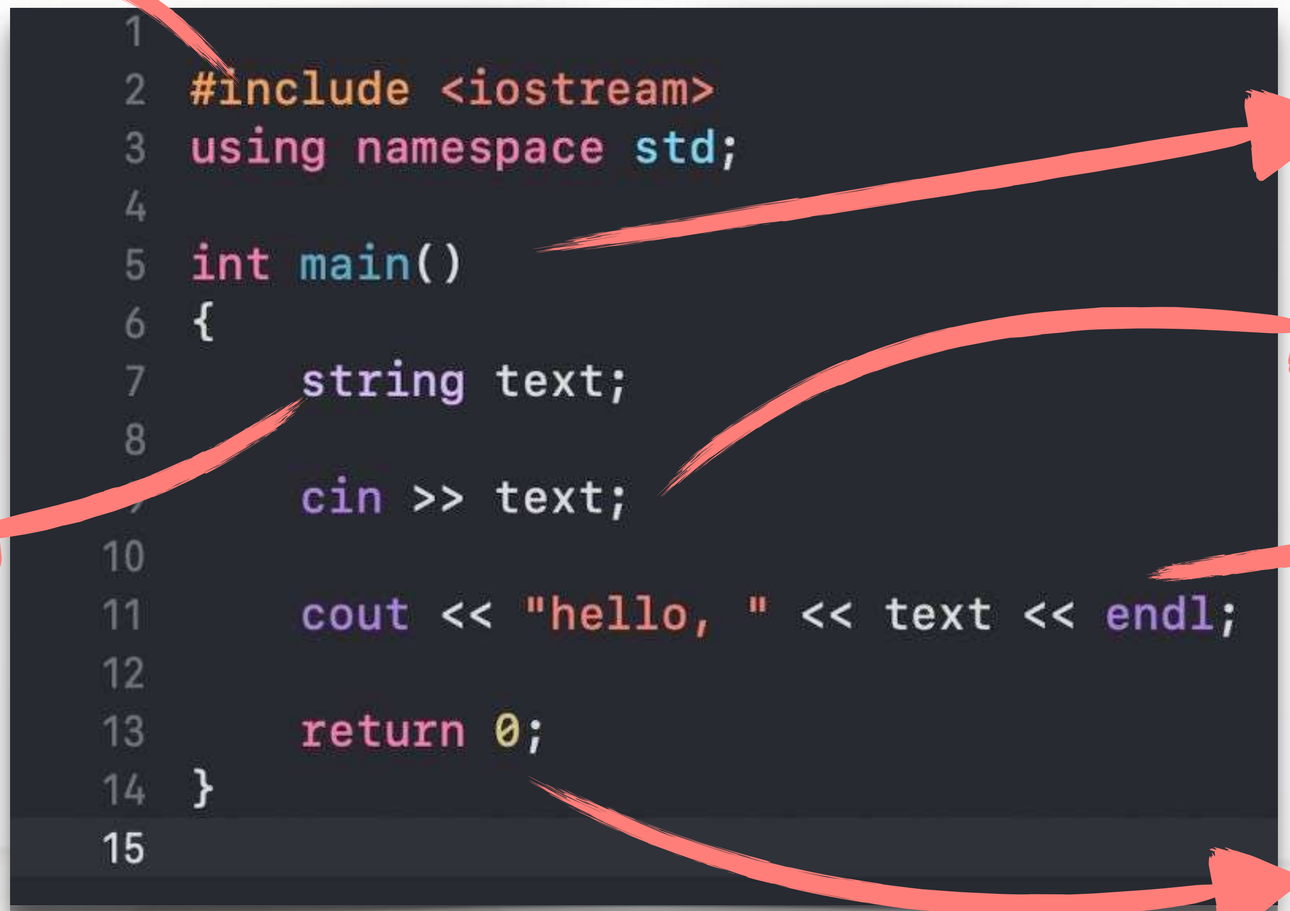
為何要選這門課？

- 我認為在未來會碰到更多與機器人相關的科技，各行各業都會與機器人相關。我不想只是會使用機器，而是了解其背後原理，並且**成為自己的一項技術**，所以想要藉這個機會**學習機器人的原理**以及**如何撰寫出讓機器人運行的程式**


學習C++入門概念

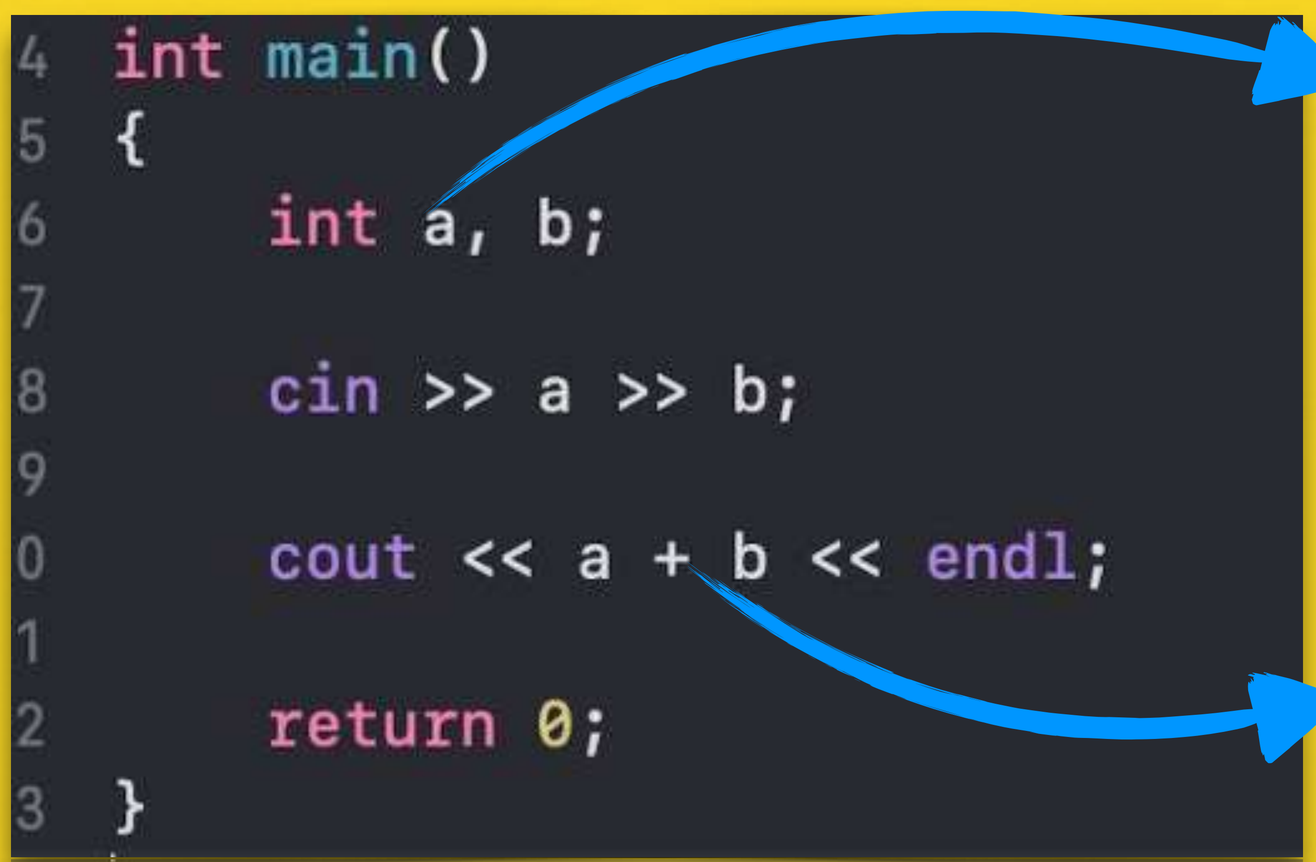
所有程式語言的第一個練習題
“Hello, world!”

- #include <iostream> 和 using namespace std是要開始**攥寫C++的標準流程**
- **”string”**是一個變數型態，用來儲存**「字串」**



- int main( ) 像是一本書的**封面**，大括弧內就是這本書的**內容**
- cin 代表**輸入**
  cout 代表**輸出**
- 讓程式**換行**
- 每一行後面要有分號”;”，代表**一件事的結束**
- return 0 是程式執行到這裡時會**結束**

學會簡單的加減運算



- 和”string”一樣的概念，差別在”int”是儲存**「整數」**
- 這裡可以依照要做什麼運算使用：**加“ + ”，減“ - ”，乘“ * ”，除“ / ”**

小節檢討

- 剛開始，會不習慣“#include <iostream> ”和“ using namespace std”，記不清楚這是做什麼用的，但後來把他**想成作文一開始要的空格**比較好記著
- 常常忘記一行後面**要有“ ; ”**，以及最後要有**“ return 0 ”**
- 通常大括號內的東西會**利用縮排“ tab ”**，讓自己能清楚判斷這是屬於哪個的

“if , else”條件式

- “ if ”後面的小括號是**條件**，大括號是條件程式時**要做的事**
- 當前面的條件**不成立**時，如果條件二成立做什麼
- 所有條件**都不成立時**要做什麼

```
int grade;

cin >> grade;

if(grade >= 80)
{
    cout << "excellent!" << endl;
}

else if(grade >=60)
{
    cout << "good!" << endl;
}

else
{
    cout << "cheer up!" << endl;
}
```

- “ >= ”就是大於等於，和平常數學用法一樣但比較**特別的是“ == ”**，用於比較同等值的時候使用，並**不是用“ = ”**

while迴圈

- 只要符合 i <= 10 那麼我的程式就能**持續執行**，直到 i = 11 那他就會**跳出**這個迴圈
- 最後執行結果就會是**1~10的數列**

```
int i;

i = 0;

while( i <= 10 )
{
    cout << i << " ";

    i = i + 1;
}
}
```

- 這裡的“ i ”會被“ i + 1 ”**替代**掉
- 像這裡，**原本 i = 0** ，第一次執行到這裡時就會變成 1，**最後變成11**

利用break跳出迴圈

- 這裡while重複的條件是1但這樣他會**一直持續**下去
- **透過break**就可以**強行跳出**迴圈，所以**i = 11** 時這個程式就**會結束**
- 結果出現**1~11的數列**

```
while(1)
{
        cout << i << " ";
        if(i > 10)
        {
                break;
        }

        i = i + 1;
}
}
```

陣列

```
int money_1 = 100;
int money_2 = 200;
int money_3 = 300;
int money_4 = 600;
int money_5 = 500;
```

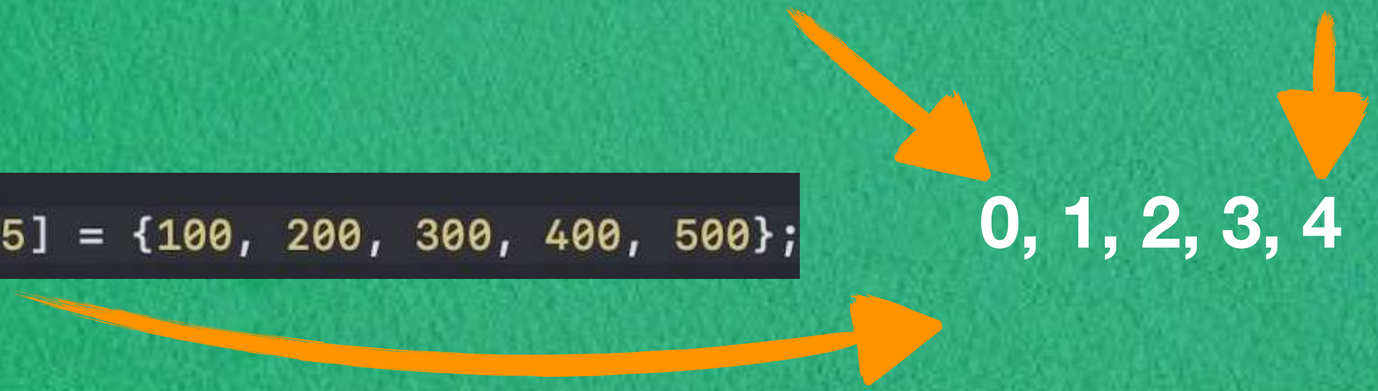
```
int money[5] = {100, 200, 300, 400, 500};
```

- 陣列可以讓我不像原本那樣**一個一個去設變數**，接著輸入數字
- 這讓我在設變數時能**減少很多時間**

小節檢討

- 已經開始使用許多的條件運算，在使用時真的要注意「縮排」這件事，這讓我在修改程式時能夠看清楚是屬於哪個條件的。
- 在使用陣列時，要記住是從**「零」**開始，直到**設定數字減1**為止。

```
int money[5] = {100, 200, 300, 400, 500};
```



驗證學習

- 學了前面這些概念老師出了一道題來驗證我們**是否有學進去**並且**能使用**

1. 要能新增客戶

2. 自行設置十名的客戶資料

資料包括：帳戶、密碼、帳戶金額、存提款記錄

3. 要能提款 / 存款 / 轉帳 /

學習過程

- 想成**實際操作atm**的狀況

- 按鈕一能做什麼...
- 按鈕二能做什麼...
- 照這個想法，我先將按鈕的作用打出來

- 要記得**換行**，不然全部會變同一行，這很重要

製作選項內容

- 先做兩個人的資料讓我**測試**程式能不能成功執行

```
name[0] = "John";
ID[0] = 1234567;
password[0] = 123;
balance[0] = 10000;

name[1] = "Mary";
ID[1] = 2345678;
password[0] = 456;
balance[1] = 9999;
```

- 利用前面學到的**陣列**來**儲存資料**

- 利用“ while ”和“ cin ”達成重複輸入

```
while(cin >> option)
{
    if(option == 1)
    {
        cout << "Enter a money to withdraw money from your ID: ";
        cin >> withdraw ;

        balance[0] = balance[0] - withdraw;
        cout << "Now your balance: " << balance[0] << endl;
    }

    else if(option == 2)
    {
        cout << "Enter a money to deposit money into your ID: ";
        cin >> deposit;

        balance[0] = balance[0] + deposit;
        cout << "Now your balance: " << balance[0] << endl;
    }

    }
```

- 利用“ if,else ”讓程式**判斷**“ 1 ”要做什麼，“ 2 ”要做什麼...

遇到問題我如何解決？

- 重新複習老師曾教過的概念，檢視自己是否有理解錯誤，若沒有理解錯誤那可能是思考方向錯了，所以會再重新思考方向

如何讓客戶登出？

- 輸入“ 5 ”時執行

```
else if (option_2 == 5)
{
    cout << " Thank you! " << endl;
    cout << "MENU " << endl;
    cout << "1...Add a new client." << endl;
    cout << "2...To withdraw, deposit, transfer,or check balance." <<
    cout << "Please enter your option: ";
    break;
}
```

- 想起之前有學過“ break ”這個概念，
  所以在這邊就使用上了

- 因為強制跳出迴圈後，會回到選擇的地方，但不會重複出現“ MENU ”
  所以得再打一次

如何讓程式判斷輸入的帳號／密碼存不存在？

- 將陣列裡的數字**改成變數“ x ”**就能實現**判斷輸入的帳號是否存在**這一千個帳號空間內了

- 利用前面所學的**“ while ”**這樣就能做一千次

```
while(x < 1000)
{
    if(ID_1 == ID[x] and password_1==password[x])
    {
        cout << "Hello " << name[x] << endl;
        cout << "1...To withdraw money from your ID." << endl;
        cout << "2...To deposit money into your ID." << endl;
        cout << "3...To transfer money to another ID." << endl;
        cout << "4...To check your balance." << endl;
        cout << "5...To exit." << endl;
        cout << "Please enter your option: ";
    }
    x = x + 1;
}
```

- 思考出這個作法後，**轉帳**的問題也以同樣做法解決了

如何儲存客戶的資料？

- 利用“ if ”只能執行一次的特性，輸入名字 / 帳號 / 密碼時只會做一次

```
z = 10;

cout << "Enter your name: " << endl;
cout << "Enter your ID: " << endl;
cout << "Enter your password: " << endl;
if(cin >> name_0 >> ID_0 >> password_0)
{
    name[z] = name_0;
    ID[z] = ID_0;
    password[z] = password_0;
    cout << "Nice to meet you " << name[z] << endl;
    cout << "Your ID: " << ID[z] << endl;
    cout << "Your password: " << password[z] << endl;

    z = z + 1;
}
```

- 同樣將陣列裡的數字改成變數“ z ”
- 最後讓變數“ z ”加一，讓這一段程式能持續新增帳戶，但是只有辦法增加990個

成果

- 成功地執行**新增帳戶功能**

```
MENU
1...Add a new client.
2...To withdraw, deposit, transfer,or check balance.
Please enter your option: 1
Enter your name:
Enter your ID:
Enter your password:
Eason
10930159
10917
Nice to meet you Eason
Your ID: 10930159
Your password: 10917
Enter 1 to withdraw, deposit, transfer,or check balance.
Enter 2 to the menu.
Please enter your option: 2
Thank you!
MENU
1...Add a new client.
2...To withdraw, deposit, transfer,or check balance.
Please enter your option:
```

帳號與密碼

- 成功的登錄帳號和密碼並執行提款、**存款**、**轉帳**、**看餘額**以及登出

```
MENU
1...Add a new client.
2...To withdraw, deposit, transfer,or check balance.
Please enter your option: 2
Enter your ID and password:
1234567 123
Hello John
1...To withdraw money from your ID.
2...To deposit money into your ID.
3...To transfer money to another ID.
4...To check your balance.
5...To exit.
Please enter your option: 1
Enter a money to withdraw money from your ID: 200
Now your balance: 9800
2
Enter a money to deposit money into your ID: 500
Now your balance: 10300
3
Enter a money and an ID to transfer money to another ID: 1000 2345678
Now your balance: 9300
Another ID balance: 10999
4
Now your balance: 9300
5
 Thank you!
1...Add a new client.
2...To withdraw, deposit, transfer,or check balance.
Please enter your option:
```

心得

- 我的的程式雖然成功，但我花了好多時間**重新理解**老師教過的**概念**
- 我發現**雖然老師教過並且做過題目**，但是我當時並**沒有真正能活用**學過的概念
- 在寫程式的過程，會碰到許多問題，每當碰到問題時，我就開了另一個專案，來單獨解決這個問題，**盡量自己思考**出來，真的不會再去查詢資料或詢問老師
- 遇到**問題**並且靠著**不斷嘗試**最後**想出解決辦法**，讓我感到滿滿的**成就感**

成長與省思

- 在atm這道題目中，我將題目**想成實際操作**的狀況，讓我在做程式時，想法上比較不會打結
- 因為我打的程式有**許多的“ if ”和“ while ”**，層數疊加的太多**導致**我想要修改一個部分時就會需要**修改其他地方**
- 要**加上註解**，除錯時比較清楚

- 在打程式時沒標示清楚功能

```
if(ID_1 == ID[x] and password_1==password[x])//檢查帳號和密碼
```

- 老師說層數是其次，能夠**學會運用**在這門課裡學到的知識並活用才是重點！
- 畢竟有需多概念我還沒有學到，所以未來繼續學習後用新的概念可能會更好