C++基礎語法

鄭詠堯

偏鄉

課程簡介

C++是一個歷史悠久的語言，於1980年代被發明，是C語言增加物件導向特性的產物。而隨著時間的演進，C++也獲得了更多更多的特性，且在運行效率上高過於很多語言(摁，python我就是在說你)。當你們接下來開始接觸競賽之後，主要就是以C++作為編寫時的語言，所以這四周的課也可以視為接下來演算法課的前導。有不懂的問題，歡迎直接問我或其他學長喔喔喔！

IDE

IDE(Integrated Development Environment)，是將程式編寫、編譯、執行結合的平台。

在編寫程式碼時幾乎都會用到IDE(有方法可以不用IDE，但…你們先學會IDE就好ㄌ)

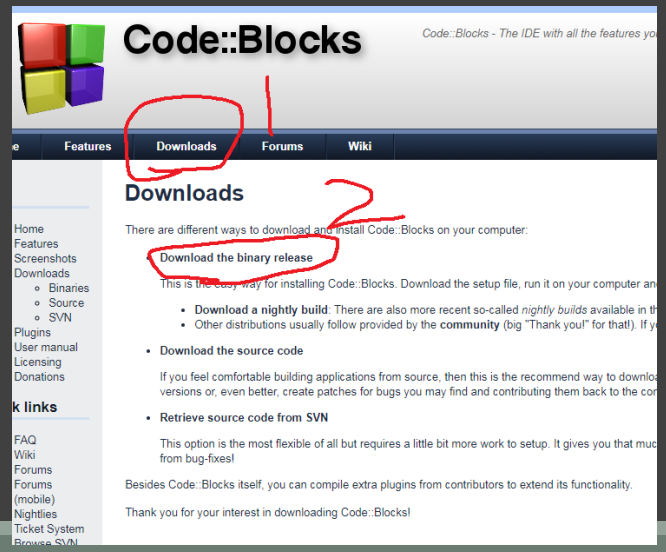
常見的C++ IDE有 VS Code, Dev C++, Code::blocks, 線上的repl.it等

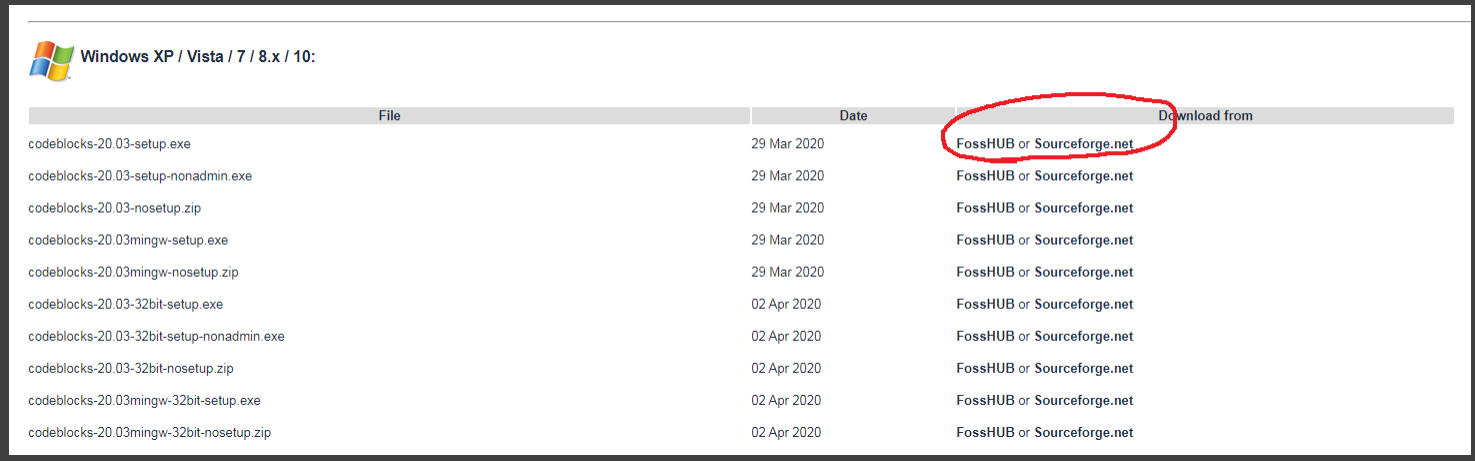
這些IDE各有優劣，那我們的主要會使用Code::blocks為主

Code::blocks下載（學校電腦已經幫你載好了喔喔喔）

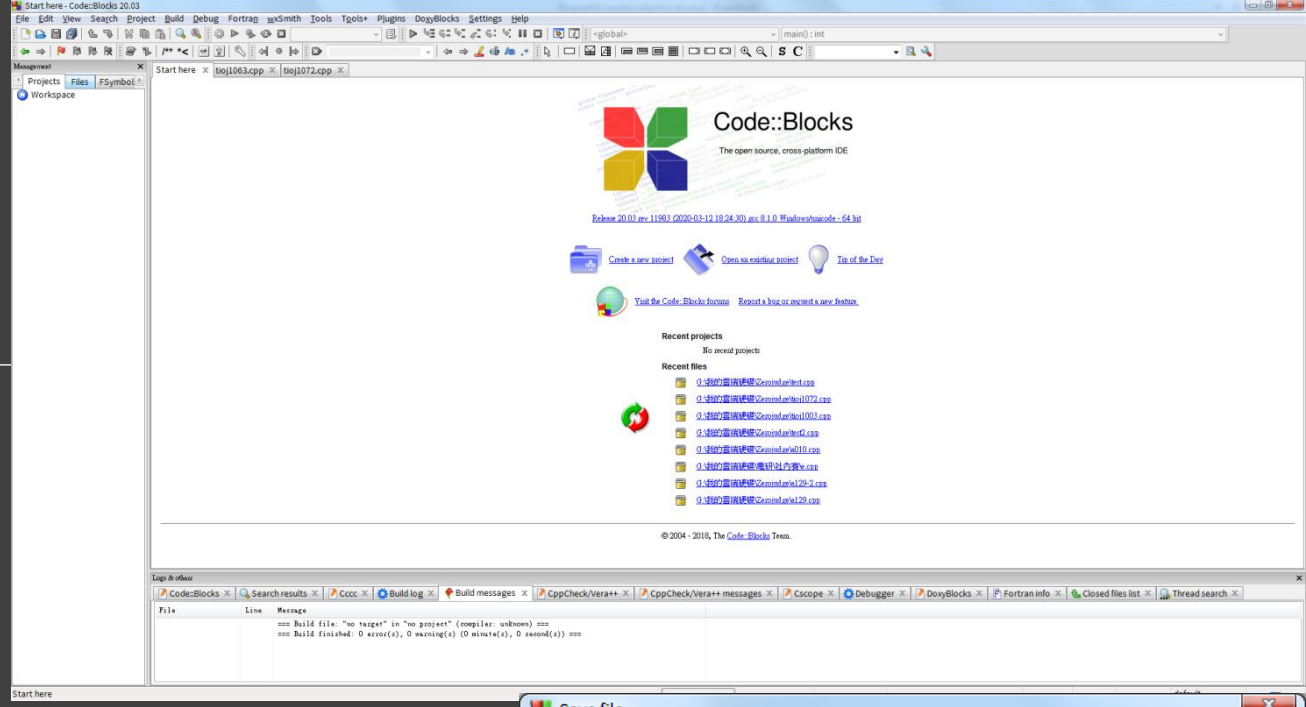
1.下載codeblocks，前往 www.codeblocks.org 並點選Downloads 及 Download the binary realease，然後再選擇最新版本的setup.exe並下載(如果電腦是32位元的則下載32bit-setup.exe)

2.打開exe檔後就按照安裝檔的指示安裝就可以了





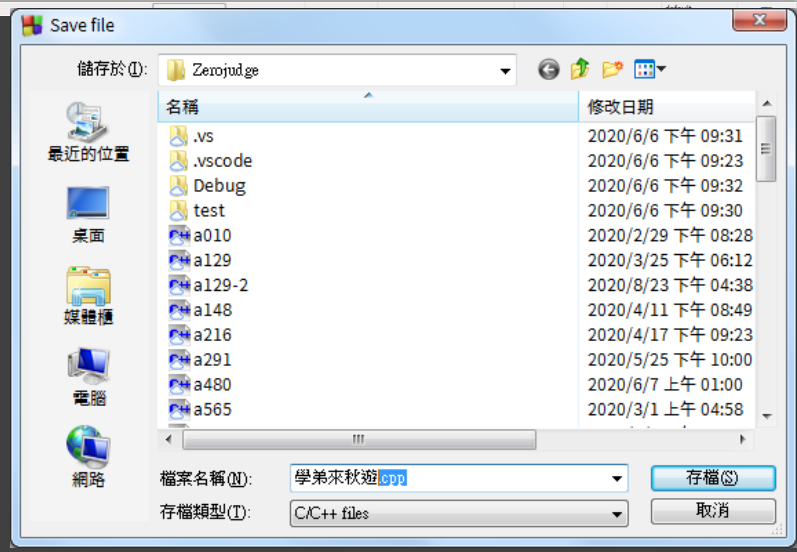
Code::blocks



右邊的是進去Code::blocks的第一個畫面，

以下是一些常用的快捷鍵。
Ctrl+Shift+N: 建立新檔案

Ctrl+S: 儲存檔案，好孩子要記得存檔喔



由於IDE通常會對特定檔案類型提供好用的功能，所以開啟新檔案後要先存檔會比較方便 //這個**.cpp超級重要**，否則codeblocks會把檔案視為c語言，所以一定要打

F9: 編譯並執行，最重要的一個快捷鍵，當你的code打完後就要按這個鍵，讓IDE把你的code轉為電腦可以執行的exe檔，並執行

每段程式碼都會有的東西

在初學的階段編寫C++時，一定會需要包含以下這段程式碼

```
using namespace std; //表示當有函式名稱衝突時使用std空間（有點複雜不過先記得打這行就好了）

int main(){ //main function，也就是主程式的意思，這裡面的東西是會被電腦執行的
    //你目前寫的code幾乎都會在這裡

    return 0; //告訴電腦程式結束了
}//喔對在//後面的東西是"註解"，只是方便記錄這段code是做什麼用的而已，不會被編譯到
```

在現在的階段，只要記得一律都先打上這些東西就好了，

變數Variable

變數是什麼呢？變數就是用來儲存資料的單位，透過宣告不同型態的變數與函式的搭配，就會形成一個可以運作的程式，而變數有分不同的種類，像是下方圖中寫的那些。

型態	中文意思	英文字義	可儲存的資料
`int`	整數	Integer	`100` 、 `-5` 、 `1246` ...
`float`	浮點數(小數)	floating point	`3.14159` 、 `4.3` 、 `-1.1` ...
`char`	字元(半形字)	Character	`'a'` 、 `'R'` 、 `'1'` 、 `'@'` 、 `'*'` ...
`string`	字串(文句)	String	`"Hello"` 、 `"^_^"` 、 `"Rock!"` ...
`bool`	布林(是非)	boolean	`true` 、 `false`

變數

而一個變數必須要先”宣告”才能去使用，宣告方式則是 型態 變數名稱； 看不懂沒關係，看以下的示範就好ㄌ

```
using namespace std;

int main(){
    int a;      //宣告整數型態變數a
    float b;    //宣告浮點數型態變數b
    string c;   //宣告字串型態變數c

    int x1,x2,x3; //你也可以一次宣告多個同型態的變數，之間用,分隔

    return 0;
}
```

要注意一點是變數的名稱可包含數字、大小寫字母、底線，但不能以數字開頭，且大小寫視為不同的東西

那現在你已經讓電腦知道了變數的存在了，要怎麼進行操作呢？

I/O

也就是Input/Output的意思，在C++中有兩種方式可以進行I/O，分別是cin/cout及scanf/printf，搭配的標頭檔則是iostream及stdio.h，那我們主要會學習cin/cout的作法

要注意一點是cin/cout是一個函式，所以要先將他們所屬的函式庫包含才能使用，作法是main函式之前加入一行`#include <iostream>`，像下面這樣

```
using namespace std;
#include <iostream> //包含函式庫iostream

int main(){
```

輸入

輸入是透過cin這個函式達成的，而cin的使用方式是`cin >>` 儲存輸入的變數，且一次可以輸入不止一個東西，每個東西間用>>間隔。那就直接看code吧。

```
using namespace std;
#include <iostream> //包含函式庫iostream

int main(){
    int x,y,z; //宣告整數變數x,y,z
    cin >> x; //使x的值變為輸入的值
    cin >> y >> z; //使y的值便為第二個輸入的值，z的質變為第三個輸入的值

    return 0;
}
```

另外cin會自動偵測，當按下enter或輸入有空白時便會視為一段輸入結束，並將輸入的東西放入變數中。

輸出

輸出是透過cout這個函式達成的，寫起來會是`cout << 要輸出的東西;`，且一次可以輸出不止一個東西，每個東西間用<<間隔。而這個輸出的內容可以是任何東西，可以是一個變數，也可以是一個整數、字串、浮點數etc。

以下是所有程式設計師的第一步，試著寫出一段可以輸出hello world的程式碼吧。

```
using namespace std;
#include <iostream> //包含函式庫iostream

int main(){
    cout << "Hello World!" << endl; //用"包住字串內容"，endl則表示換行符號

    return 0;
}
```

題目練習看看

去寫寫看zerojudge d483跟zerojudge a001，我懶得出題ㄌ

運算子

運算子就是對變數進行操作的方式，有分為指派、算數、算術指派、邏輯、關係、位元等種類的運算子，前5個是你們短期之內比較常用的，我們也會先學習這幾個。

運算子列表 維基百科有列出完整的表喔，包含一些等一下不會教到的，有空的話可以看看，不看也沒關係，之後有機會再學就好ㄌ。

指派運算子

也就是將一個變數的值指定為某個值，例如x = 0;。而除了將變數指定為一個值，也可以讓變數的值等於另一個變數，例如x = y;

```
using namespace std;
#include <iostream>

int main(){
    int x,y; //宣告變數x,y
    x = 1;   //將x的值指定為1
    y = 2;   //將y的值指定為2
    cout << x << ' ' << y << endl; //輸出x的值及y的值

    x = y;   //將x的值指定為y的值
    cout << x << ' ' << y << endl; //輸出x的值及y的值

    return 0;
}
```

這樣第一次輸出的值就會是1 2，而第二次由於x的值被指定為y的值，因此便是輸出2 2。

初始值

另外，也可以在宣告變數時先指派初始值，像下面那樣喔。跟前面先宣告完再賦值是一樣的意思。

```
int main(){
    int x = 1,y = 2; //宣告變數x,y，並將x初始值設為1，y的初始值設為2
```

算數運算子

主要是加減乘除及模(取餘數)這五項，在C++中分別對應的是 +,-,*,/,% 的符號，而他們的作用就與數學中相似，且通常會跟指派一起使用。另外，跟數學一樣，用括號包住的東西

```
using namespace std;
#include <iostream>

int main(){
    int x = 1,y = 2; //宣告變數x,y，並將x初始值設為1，y的初始值設為2
    int a,b,c,d,e,f;   //宣告整數變數a,b,c,d,e,f

    a = x + y;  //指派a的值為x+y
    b = x - 10; //指派b的值為x-10
    c = x * y;  //指派c的值為x*y
    d = x / y;  //指派d的值為x/y
    e = 31 % y; //指派e的值為31取2的餘數
    f = (x + 14) / (y + 1); //指派f的值為(x + 14)除以(y+1)

    cout << a << ' ' << b << ' ' << c << ' ' << d << ' ' << e << ' ' << f << endl;
    //輸出a,b,c,d,e,f的值

    return 0;
}
```

另外，也可以寫成`a = a + 5;`的語法，這樣會使a的值變為`a + 5`喔

算術指派運算子

在前一頁有提到的a = a + 5的語法，可以被簡化為a += 5;，而-,*,/,%也都可以用一樣的方式喔。

而如果是要加1或減1的話，那可以再更簡化為++或--的用法喔。

```cpp
using namespace std;
#include <iostream>

int main(){
    int a = 1,b = 1,c = 1,d = 1; //宣告整數變數a,b,c,d，並將初始值都設為1
    a += 10; //a的值增加10
    b -= 10; //b的值減少10
    c *= 10; //c的值乘以10
    d /= 10; //d的值除以10

    a++; //等同於a += 1
    ++b; //反過來也可以喔
    c--; //等同於c -= 1
    --d; //反過來還是可以喔

    cout << a << ' ' << b << ' ' << c << ' ' << d;  //輸出a,b,c,d

    return 0;
}
```

Try踹看

就...踹踹看，可以寫個zj002，自己玩玩看也可以啦。

If else

在進入剩下兩種運算子前，我們先來看看他們ㄉ主要使用地方吧。也就是if else，而它們便是能使程式判斷不同條件的方法。

而程式碼實作的部分則會是像是下面這樣

```
if (條件式){
    //執行內容
}
```

選擇結構

而執行內容可以是前面教過的那些東西，或者是將來會學到的更多函式等等的喔。條件式則是會回傳一個值，如果這個值可以表達`true`(也就是所有非`0`的值`1`)，便執行後面大括號內的內容。

`if`是一個選擇結構裡面一定有的東西，可以單獨出現，也可以後面接東西。

`else if`則是要接在一個`if`或另一個`else if`後面，如果前面的`if`為`false`才會判斷到，否則直接跳過。另外`else if`可以放不只一個。

`else`則沒有條件式，是當前面的`if, else if`條件式都沒有通過，才會執行內容。

關係運算子

那前面提到的判斷式要怎麼建構呢？通常都是透過更前面提到的關係運算子來達成。關係運算子跟數學上的等式、不等式有點相似，也就是表達左式與右式的關係，若關係為真，回傳前面提過的布林值`true`，為非則回傳布林值`false`。

關係運算子的種類

== :如果兩邊相等便回傳true

!= :如果兩邊不相等便回傳true

> :如果左式大於右式便回傳true

< :如果左式小於右式便回傳true

>= :如果左式大於等於右式便回傳true

<= :如果左式小於等於右式便回傳true

```cpp
using namespace std;
#include <iostream>

int main(){
    int a = 1,b = 2,c = 3,d = 4; //宣告整數變數a,b,c,d，並將初始值分別設為1,2,3,4
    if (b > a){
        cout << "b>a";
    }
    else if (b > a){
        cout << "這邊不會被執行到喔!";
    }

    if (a == b){
        cout << "不等於喔!";
    }
    else{
        cout << "if(a==b)為非所以執行這裡!"
    }

    return 0;
}
```

邏輯運算子

而如果同時有兩個條件必須通過怎麼辦，或者有兩個條件都可以，這時就要透過邏輯運算子去達成。而邏輯運算子有三種分別是OR,AND,NOT，OR跟AND會連接兩個判斷式，當兩判斷式成特定關係時，便回傳`true`。

OR　:運算子寫成 || ，只要有一邊為`true`，便回傳`true`

AND :運算子寫成&&，若兩邊都為`true`，則回傳`true`

NOT :運算子寫成！，只會接在一個式子前面，如果那個式子是`true`，則回傳`false`，反之亦然

範例程式碼

前一頁放不下ㄌ

```
using namespace std;
#include <iostream>

int main(){
    int a = 1,b = 2,c = 3,d = 4;
    if (a > b || !(d > c)){
        cout << "!(d > c) 跟 c <=d 意思一樣喔"
    }
    else if (b > a && c > b){
        cout << "c > b > a";
    }

    return 0;
}
```

寫寫題目

zj a003

下禮拜要教ㄌ

迴圈跟陣列~

記得來ㄛ