

105學年 資料結構TA課

運算子多載(operator overloading)

Outline

- C++的運算子
 - 算數運算子(Arithmetic operators)
 - 比較運算子(Comparison operators)
 - 邏輯運算子(Logical operators)
- 運算子多載

C++運算子

a = 5
b = 2

算數運算子

運算子名稱	語法	輸出結果
等於(賦值)	a = b	a = 2 b = 2
加	a + b	7
減	a - b	3
乘	a * b	10
除	a / b	2.5
取餘數	a % b	1
遞增	++a 或 a++	a = 6
遞減	--a 或 a--	a = 4

已遞增為例

遞增 × 遞減

++a 和 a++ 有何不一樣？不都一樣是 $a=a+1$ ；

答：結果一樣但是中間過程不太一樣

會先加1再輸出

```
int A1=10, A2=10;
```

```
cout << "++A1 => " << ++A1 << endl;
```

```
cout << "A1 = " << A1 << endl;
```

```
cout << "A2++ => " << A2++ << endl;
```

```
cout << "A2 = " << A2 << endl;
```

++A1 => 11

A1 = 11

A2++ => 10

A2 = 11

會先輸出再加1

a = 5
b = 2

比較運算子

運算子名稱	語法	輸出結果
是否等於	a == b	FALSE
是否不等於	a != b	TRUE
大於	a > b	TRUE
大於等於	a >= b	TRUE
小於	a < b	FALSE
小於等於	a <= b	FALSE

a = true
b = false

邏輯運算子

運算子名稱	語法	輸出結果
NOT	!a	FALSE
AND	a && b 或 a and b	FALSE
OR	a b 或 a or b	TRUE

運算子多載

運算子多載

- 在C++中是可以對基本的型態進行運算子的運算，例如int、double...
- 但是有時會想將兩個物件進行運算的時候C++沒有預設此方法
- 因此運算子多載就可以讓我們自定義運算子的方法

運算子多載

- 舉例來說，我們現在有一個class，這個class存放實數與虛數。

```
class number
{
private:
    int r; // 實數
    int i; // 虛數
public:
    number(int re=0, int im=0)
    {r = re; i = im;}
}
```


運算子多載

- 然後建立了a、b、sum三個物件，並且想要將a+b後的值存到sum。

```
number a (10,10);  
number b (3, 5);  
number sum;  
sum = a + b;
```

- 如果未實作出該 class 的加法多載，該行就無法使用。

運算子多載

operator overloading.cpp

```
class Number
{
private:
    int r, i;
public:
    Number(int re=0, int im=0):r(re), i(im) {};
    friend Number operator+ (const Number &, const Number &);
    void out();
};

Number operator+ (const Number &a, const Number &b)
{
    Number c;
    c.r = a.r+b.r;
    c.i = a.i+b.i;
    return c;
}

void Number::out()
{
    _
}
```


練習

- 用 class 和 operator overloading 實作出，實數與虛數的加減乘除。