

第四章運算元、算符與算式

資訊科技系

林偉川

算式結構

- 在一個算式中, 利用算符來作運算元的運算處理, 比如說加減乘除、比大小或邏輯運算 (AND, OR...) 等。算式的結構如下：

運算元1 算符 運算元2

運算元的種類

- **運算元**可以為數值或者變數,甚至可以是其他的算式。而算符可以細分成以下 3 種：
- 算數算符：為數學的基本運算,如加、減、乘、除等(+、-、*、/、%)。
- 邏輯算符：將兩個數值以邏輯關係結合,如 **AND**、**OR**、**NOT**(&&、||、!)。(& | !)
- 關係算符：描述兩個數值的關係,如大於、小於、等於(>、<、==)。

3

算式結構

- 由運算元加上算符才能算是一個完整的算式,以一個簡單的數學式子 $a+b$ 來看, $a+b$ 是算式, $+$ 符號是算符, a 、 b 則是運算元。再舉一些例子來看：

$10 > 5$ $\leftarrow 10、5$ 是運算元, $>$ 符號是算符

$(a+b)*(a-b)$ $\leftarrow (a+b)、(a-b)$ 是運算元, $*$ 符號是算符

$math + chinese + english$ $\leftarrow math、chinese、english$ 是運算元, 兩個 $+$ 符號是算符

$balance = income - output$ $\leftarrow balance、income、output$ 是運算元, $=、-$ 符號是算符

4

算符的優先權

- 當一個算式中, 出現了兩個以上的運算符號時, 會有先處理哪一個算符的問題。
- C 語言也是依『先乘除、後加減』的規律進行運算。但 C 語言的算符種類較多, 將各種不同的算符用在一個算式中時, 就要依 C 語言所賦與各算式的優先權大小, 來決定要先進行哪個部份的計算。

5

算符的優先權

優先權	算符
1(最高)	() 括號
2	+(正號)、-(負號)、!(NOT)、 ++(累加, 置於變數前)、--(累減, 置於變數前)
3	*(乘)、/(除)、%(餘數)
4	+(加)、-(減)
5	>(大於)、<(小於)、>=(大於或等於)、 <=(小於或等於)
6	=(等於)、!=(不等於)
7	&&(AND)、 (OR)
8	=(等號)
9(最低)	++(累加, 置於變數後)、--(累減, 置於變數後)

6

優先權處理原則

1. 優先權高者, 先處理：所以在算式 $a+b*c$ 中, 計算順序為 $b*c$ 後再加上 a 。
2. 優先權同者, 由左至右處理：加法算符 $+$ 與減法算符 $-$ 的優先權相同, 所以如算式 $a-b+c$ 的計算順序是 $a-b$ 後的數值再加上 c 才是算式的結果。

7

優先權處理原則

3. 括號算符 $()$ 擁有最高優先權：以算式 $(a+b)*c$ 為例, 計算順序為先算括號中的 $a+b$ 的數值, 再去乘以 c 才是這個算式的正確答案。
4. 指定算符 $=$ 擁有最低優先權：等號算符會是算式中最後被處理的算符。如算式 $a=b+c$, a 的數值會等於 $b+c$ 的結果。

8

算數算符

- 算數算符是運用最普遍的算符，這類算符總管了數值間的四則運算以及正負號的指定。算數算符的基本功能可分成 7 種，如下表：

優先權	算符	功能	例子
2	+	正號	正 5 的正值, + 常被省略, 如 +5
2	-	負號	負 10, -10
4	+	加	x 的變數值加上 y 的變數值, x+y
4	-	減	一數列的最大值減去最小值 max-min
3	*	乘	3 乘以 2, 3*2
3	/	除	10 除以 5, 10/5
3	%	求餘數	17 除以 3 的餘數, 17%3

9

加減算符 & 正負號分辨的方式

- 正負號與加減的算符符號相同，分辨的方式是它們使用的位置。
- 在運算元前面的是正負號，在兩個運算元間的是加減算符：
 - +100 or 100 ➔ 正 100
 - +a or a ➔ 正 a
 - 100 ➔ 負 101
 - 100+-101 ➔ 100 加上 負 101

10

計算本月的收支平衡

1

程式 Ch04_01.c, 計算本月的收支平衡

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int balance, income, output;
06     income= 25000;
07     output= 3500;
08
09     balance= income+ -output; /* 結餘等於收入加上支出 */
10     printf(" 本月的結餘  %d\n", balance);
11
12     return 0;
13 }
```



執行結果

本月的結餘 21500

11

加減乘除的混合運算

- 一超市的貨品售價, 蛋一打 egg=30 元、土司 1 條 toast=18 元、火腿 1 斤 ham=34 元、牛奶一瓶 milk=25 元。現在要買兩打蛋、3 條土司、半斤火腿以及 4 瓶牛奶, 總共多少錢?
- 將此例題寫成程式, 如下:

12

計算購物金額

2

程式 Ch04_02.c, 計算購物金額

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int egg=30, toast=18, ham=34, milk=25; /* 宣告各項商品的單價 */
06     int total;
07
08     total=egg*2+toast*3+ham/2+milk*4; /* 總數等於各項商品金額的和 */
09     printf(" 總共 %d 元\n",total);
10
11     return 0;
12 }
```



執行結果

總共 231 元

13

除法算符/與餘數算符%

- 在做除法運算時, 難免會出現需計算到小數才能整除, 或是根本無法整除的情形。如下, 已知 2 個浮點數 $a=7, b=2$; 另有 2 個整數 $c=7, d=2$:
 $a/b \rightarrow 3.5$ V.S. $c/d \rightarrow 3$

14

驗證浮點數相除與整數相除的結果

3

程式 Ch04_03.c, 驗證浮點數相除與整數相除的結果

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     float a=7,b=2;
06     int c=7,d=2;
07
08     printf("兩浮點數相除,結果為 %f\n",a/b);
09     printf("兩整數相除,結果為 %d\n",c/d);
10
11     return 0;
12 }
```



執行結果

兩浮點數相除, 結果為 3.500000

兩整數相除, 結果為 3

15

餘數算符%

- 如果想要求餘數,就需要用到 % 的算符。餘數算符 % 的使用方法很簡單,只要在除法運算中,以 % 代替 /,則算式所得到的數值就是餘數而非商數,只是有一點要特別注意的,使用餘數算符 % 時,除數與被除數都必須是整數型別。

16

餘數算符% 範例

- 假設今天有 100 顆球, 每次拿掉 7 個, 拿到最後不足 7 之數剩幾個。以數學的角度來看, 這個問題很簡單只要用 100 除以 7 所得到的餘數就是答案, 如果要寫成程式就必須用到餘數算符, 如下:

17

餘數算符 % 範例

4 程式 Ch04_04.c, 計算 100 顆球每次拿掉 7 個最後不足 7 的剩幾個

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int totalball;
06     int ball,left;
07     totalball=100;      /* 球的總數 100 個 */
08     ball=7;             /* 每次拿掉 7 個球 */
09
10     left=totalball % ball; /* 得到兩數相除後的餘數 */
11     printf(" 剩下 %d 個球\n",left);
12
13     return 0;
14 }
```



執行結果

剩下 2 個球

18

關係算符

- 關係算符是用來比較每個數值的大小關係，共有 6 個：

優先權	算符	功能	例子
5	>	大於於	$a > b$, a 大於 b
5	<	小於	$a < b$, a 小於 b
5	>=	大於或等於	$a >= b$, a 大於等於 b
5	<=	小於或等於	$a <= b$, a 小於等於 b
6	==	等於	$a == b$, a 等於 b
6	!=	不等於	$a != b$, a 不等於 b

19

關係算符

- 關係算符的結果只有兩種，一種是**真**；一種是**假**。
- 運算結果為**真**會以整數 1 來表示(**非0為之真**)，**假**則以整數 0 來表示。

20

關係算符

- 父親的年齡 `fatherage` 一定比小孩年齡 `childage` 大, 所以：

<code>fatherage > childage</code>	← 結果為 1, 因為這是真的
<code>fatherage < childage</code>	← 結果為 0, 因為這是假的
<code>fatherage == childage</code>	← 結果為 0, 因為這是假的
<code>fatherage != childage</code>	← 結果為 1, 因為這是真的
<code>fatherage >= childage</code>	← 結果為 1, 父親年齡大於或等於小孩年齡 只要有一個成立為真
<code>fatherage <= childage</code>	← 結果為 0, 因為父親的年齡一定不會小於 或等於小孩年齡

- 把父親的年齡明定為 38 歲, 小孩年齡定為 10 歲, 寫成程式來驗證如下：

21

判斷敘述是否正確

5 程式 Ch04_05.c, 判斷敘述是否正確

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int fatherage=38,childage=10;
06
07     printf("結果為 1 表示正確; 結果為 0 表示錯誤\n");
08     printf("父親年齡大於小孩年齡 %d\n",fatherage>childage);
09     printf("父親年齡小於小孩年齡 %d\n",fatherage<childage);
10     printf("父親年齡等於小孩年齡 %d\n",fatherage==childage);
11     printf("父親年齡不等於小孩年齡 %d\n",fatherage!=childage);
12     printf("父親年齡大於或等於小孩年齡 %d\n",fatherage>=childage);
13     printf("父親年齡小於或等於小孩年齡 %d\n",fatherage<=childage);
14
15     return 0;
16 }
```

執行結果

結果為 1 表示正確; 結果為 0 表示錯誤
父親年齡大於小孩年齡 1
父親年齡小於小孩年齡 0
父親年齡等於小孩年齡 0
父親年齡不等於小孩年齡 1
父親年齡大於或等於小孩年齡 1
父親年齡小於或等於小孩年齡 0

22

關係算符

- 有一點要特別注意的是, 關係算符可以用於比較不同資料型別的數值。比如說 float num1=10.01, int num2=10則num1 > num2為真。如果 float num1=10.00 則 a 與 b 相等。
- 如果是字元資料型別, 則會先轉成相對應的 ASCII 碼數字後, 再比較大小。

23

邏輯算符的意義與推演

- 在邏輯算符中, 把所有的運算元都歸類成兩種: 如果運算元的值為非 0 值則為真; 如果運算元的值為 0 則為假。所以邏輯算符可以說用來討論真假之間的 3 種邏輯關係, 如下:

優先權	算符	名稱
7	&&	AND (而且)
7		OR (或)
2	!	NOT (非)

24

邏輯算符的意義與推演

- 邏輯算符是取運算元的真假值來參與運算，運算結果也只有兩種，真與假。數學中的邏輯推演表如下：

AND	1	0	OR	1	0	NOT	1	0
1	1	0	1	1	1	0	1	
0	0	0	0	1	0			

25

邏輯算符與其他算符的混合運用

- 以關係算符的算式當成是邏輯算符的運算元，則會以算式結果來判斷真假，關係算符的算式結果為1則歸類為真，結果為0則歸類為假。比如說，已知 $3>2$ 和 $2>1$ 是真，根據邏輯推演表可以推演出幾個結果：(請注意優先權)

$3>2 \ \&\& \ 2>1$	← 結果為 1，因為 $3>2(1) \ \text{AND} \ 2>1(1)$ 為真
$3>2 \ \&\& \ 1<2$	← 結果為 0，因為 $3>2(1) \ \text{AND} \ 1<2(0)$ 為假
$3>2 \ \ 1<2$	← 結果為 1，因為 $3>2(1) \ \text{OR} \ 1<2(0)$ 為真
$!(3>2)$	← 結果為 0，因為 $3>2$ 為真，非真就是假

26

邏輯算符與其他算符的混合運用

- 是用算數算符的算式來當成邏輯算符的運算元，當算式結果為 0 則會被邏輯算符當成假來處理，如果是非 0 的結果則會當成真來處理，如 $a=2, b=2$ 則：

$a+b \ \&\& \ a-b$ ← 結果為 0，因為 $a+b$ 為 4 是真， $a-b$ 為 0 是假，『真 AND 假』為假

$a+b \ || \ a-b$ ← 結果為 1，因為『真 OR 假』為真

$!(a+b)$ ← 結果為 0，因為非真為假

27

從螢幕輸出邏輯推演的結果

6 程式 Ch04_06.c 從螢幕輸出邏輯推演的結果

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int a=2,b=2;
06
07     printf("a+b>a-b && a+b<a-b 的邏輯推演結果為 %d\n",
08           a+b>a-b && a+b<a-b);
09     printf("a+b>a-b || a+b<a-b 的邏輯推演結果為 %d\n",
10           a+b>a-b || a+b<a-b);
11
12     return 0;
13 }
```



執行結果

$a+b>a-b \ \&\& \ a+b<a-b$ 的邏輯推演結果為 0
 $a+b>a-b \ || \ a+b<a-b$ 的邏輯推演結果為 1 28

其他算符

- 指定(=)算符
- 累加 ++ 與累減 -- 算符
- 在 C 語言中, 關係算符的等於符號為 == ; 指定的符號是 = 。

29

利用 = 符號將各種不同的值 指定給變數, 求最後的值

7

程式 Ch04_07.c, 利用 = 符號將各種不同的值指定給變數, 求最後的值

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int i;
06     int j;
07     i = 10;          /* i 的值等於 10 */
08
09     j = i + 10;      /* j 的值等於 10 + 10 */
10     i = i + j;        /* i 的舊值 10 加上 j 的值 20 */
11                     /* 變成了新的 i 值 30 */
12     j = j + 1;
13     printf(" i 的值為 %d\n", i);
14     printf(" j 的值為 %d\n", j);
15
16     return 0;
17 }
```



執行結果

i 的值為 30
j 的值為 21

30

等號與其它算符合併

- =算符也可與算數算符合併使用，將原來的算式簡寫，如下表：

算符	原來的算式	縮減後
+=	$a=a+b$	$a+=b$
<=	$a=a-b$	$a-=b$
*=	$a=a*b$	$a*=b$
/=	$a=a/b$	$a/=b$
%=	$a=a\%b$	$a\%=b$

31

計算 $a=a+b$ 與 $a=a*b$ 的值

8

程式 Ch04_08.c, 計算 $a=a+b$ 與 $a=a*b$ 的值

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int a=23, b=51;
06
07     a+=b; /* 也就是 a = a + b */
08     printf("a = %d\n",a);
09     a*=b; /* 也就是 a = a * b */
10     printf("a = %d\n",a);
11
12     return 0;
13 }
```

執行結果

a = 74

a = 3774 ←最後 a 的值已經不是 23，而是 74 乘以 51

32

累加 ++ 與累減 -- 算符

- ++ 的算符可以使變數的值加 1, -- 的算符可以使變數的值減 1, 其使用法如下表所示：

優先權	算符	原來的算式	縮減後
2	++ (置於變數前)	$a = a + 1$	$++a$
2	-- (置於變數前)	$a = a - 1$	$--a$
9	++ (置於變數後)	$a = a + 1$	$a++$
9	-- (置於變數後)	$a = a - 1$	$a--$

33

累加 ++ 與累減 -- 算符

- ++與--算符可以放在變數前面或者後面, 雖然兩者的最後結果都相同, 但是放在含有加減乘除等其他算符的算式中, 則算式執行前其數值卻是不同。像 $a++$ 表示先用 a 的值運算之後再加 1; 而 $++a$ 則是用加 1 之後的 a 值也就是 $a+1$ 的值來運算。

34

使用++a代替a=a+1

9 程式 Ch04_09.c, 使用 ++a 代替 a=a+1

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int a=23;
06
07     printf(" ++a*10 = %d\n",++a*10);
08     printf(" a= %d\n",a);
09
10     return 0;
11 }
```



執行結果

```
++a*10 = 240
a= 24
```

35

計算a=a+1的結果

10 程式 Ch04_10.c, 計算 a=a+1 的結果

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int a=23;
06
07     printf(" a++*10 = %d\n",a++*10);
08     printf(" a= %d\n",a);
09
10     return 0;
11 }
```



執行結果

```
a++*10=230
a= 24
```

36

錯誤的累加、累減語法

- 利用累加或累減配合其他算符來作運算時，請將累加、累減算符與其他算符以空格空開或加上括弧。否則，編譯器將無法辨識類似以下的兩個錯誤語法：

`a=b+++++c;`

`a=b-----c;`

- 正確的語法如下：`b=5,c=6`
`a=b++ ++c; or a=(b++) + (++c);`
`a=b-- --c; or a=(b--) - (--c);`

37

綜合演練

- 溫度度量衡的轉換：華氏轉攝氏
- 輸入半徑，計算圓形面積
- 換幣機程式
- 判斷為大寫或小寫字母

38

將華氏溫度轉成攝氏溫度

11 程式 Ch04_11.c, 將華氏溫度轉成攝氏溫度

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     float C, F;
06
07     printf("請輸入華氏的溫度\n");
08     scanf("%f",&F);
09     C = (F - 32) * 5 / 9; /* 溫度轉換公式的算式 */
10     printf("攝氏溫度為 %f\n",C);
11
12     return 0;
13 }
```



執行結果

請輸入華氏的溫度
78.8
攝氏溫度為 26.000002

39

輸入半徑, 計算圓形面積

12 程式 Ch04_12.c, 輸入半徑, 求圓形的面積

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     float area,r,pai=3.14; /* 將面積,半徑,圓周率宣告成浮點數型 */
06                             /* 別,並設定圓周率 π 的值 */
07     printf("請輸入圓的半徑\n");
08     scanf("%f",&r);
09     area = pai * r * r;
10     printf("圓的面積等於 %6.2f\n",area);
11
12     return 0;
13 }
```



執行結果

請輸入圓的半徑
2.35
圓的面積等於 17.34

40

換幣機程式

13 程式 Ch04_13.c, 模擬換幣機程式

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int money;
06     int ten,five,one;
07
08     printf(" 請輸入您的換幣金額\n");
09     scanf("%d",&money);
10     ten = money / 10;      /* 計算拾圓硬幣的個數 */
11     five = (money%10)/5;   /* 計算伍圓硬幣的個數 */
12     one = (money%10)%5;    /* 計算壹圓硬幣的個數 */
13     printf(" 零錢\n拾圓 %d 個\n五圓 %d 個\n壹圓 %d 個\n",
14           ten,five,one);
15
16     return 0;
17 }
```



執行結果

```
請輸入您的換幣金額
127
零錢
拾圓 12 個
五圓 1 個
壹圓 2 個
```

41

判斷為大寫或小寫字母

14 程式 Ch04_14.c 判斷由鍵盤輸入的字母為大寫或小寫

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int uppercase,lowercase;
06     char ch;
07
08     printf(" 請輸入一個字母\n");
09     scanf("%c",&ch);
10     uppercase = ch>64 && ch<91; /* 大寫字母判斷 */
11     lowercase = ch>96 && ch<123; /* 小寫字母判斷 */
12
13     if (uppercase==1)
14         printf(" 您輸入的是大寫字母\n");
15     if (lowercase==1)
16         printf(" 您輸入的是小寫字母\n");
17
18     return 0;
19 }
```



執行結果

```
請輸入一個字母
a
您輸入的是小寫字母

請輸入一個字母 ← 再執行一次
D
您輸入的是大寫字母
```

42

除錯

- `a+b=c+d`
`a < > b`
`a => b`
- 執行結果為何？
`int a=0,b=0,c=0;`
`a<b+4 //1`
`b==c //1`
`a+b+c == -3*-b //1`
`a && b //0`
`!b*4 || 3-3 //1`
`!c-!!b+!0 //2`

43

執行結果

- 執行結果為何？
`int b=1,c=1,a;`
`a=(++b)+(++c); //4 2 2`
`a=(b++)+(++c); //5 3 3`
`a=(++b)+(c--); //7 4 2`
`a=(b--)+(--c); //5 3 1`
- 執行結果為何？
`int a=0,b=1;`
`printf(“%d”,!a && !b); //0`
`printf(“%d”,!a || !b); //1`
`printf(“%d”,a && a && b); //0`
`printf(“%d”,a || b || b); //1`

44

執行結果

- 執行結果為何？

```
int a=4,b=3,c;  
c=(--a)+(a--);  
printf("a=%d c=%d\n",a,c);  
c=(b++) + (++b);  
printf("b=%d c=%d\n",b,c);
```

但是 當b=3, a=4;

```
c=(b++) + (b++);  
printf("b=%d c=%d\n",b,c);  
c=(a--)+(a--);  
printf("a=%d c=%d\n",a,c);
```

45

作業

- 試寫一程式輸入整數半徑，算出球體之體積
- 試寫一程式輸入三角形三邊長(a,b,c)，算出其之面積[hint: $k=(\text{週長和})/2$
面積的平方= $k*(k-a)*(k-b)*(k-c)$]
- 試寫一程式模擬捷運購票機，輸入為五十元硬幣、十元硬幣、五元硬幣各幾枚，輸出為存入之總額

46