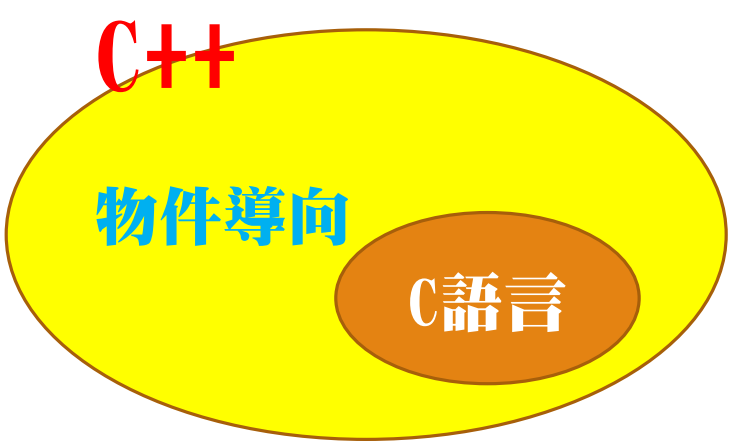
C++程式設計

單元一：概論

C++

- C++是由貝爾實驗室的比雅尼・斯特勞斯特魯普(Bjarne Stroustrup)博士以C語言為基礎加入物件導向的功能而成。



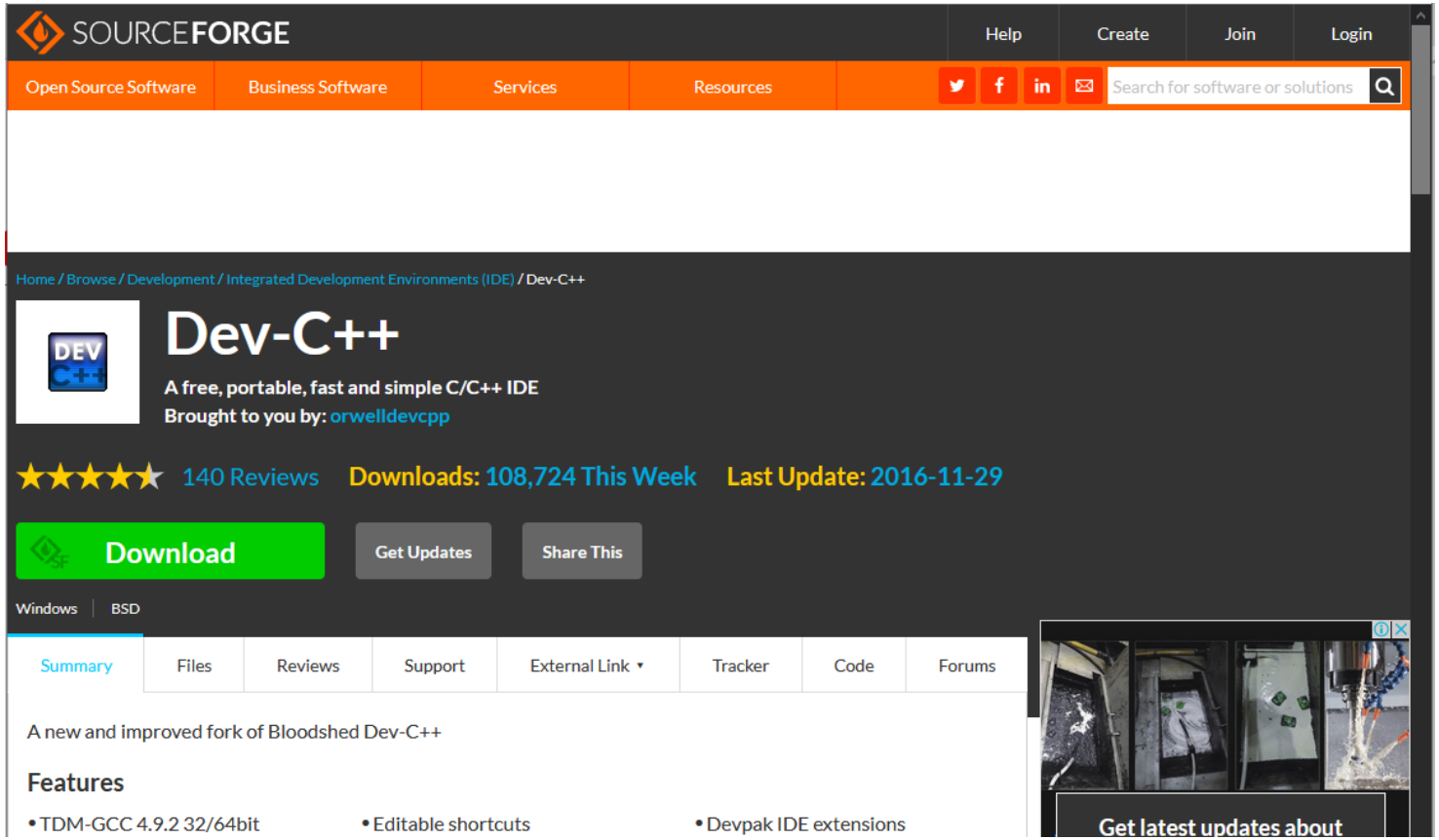
C++整合開發環境(IDE)的選擇

IDE指的是將程式編輯器、編譯器、連結器、除錯器、執行程式等功能整合在同一個軟體上，以方便程式開發，常見的C++開發環境如下：

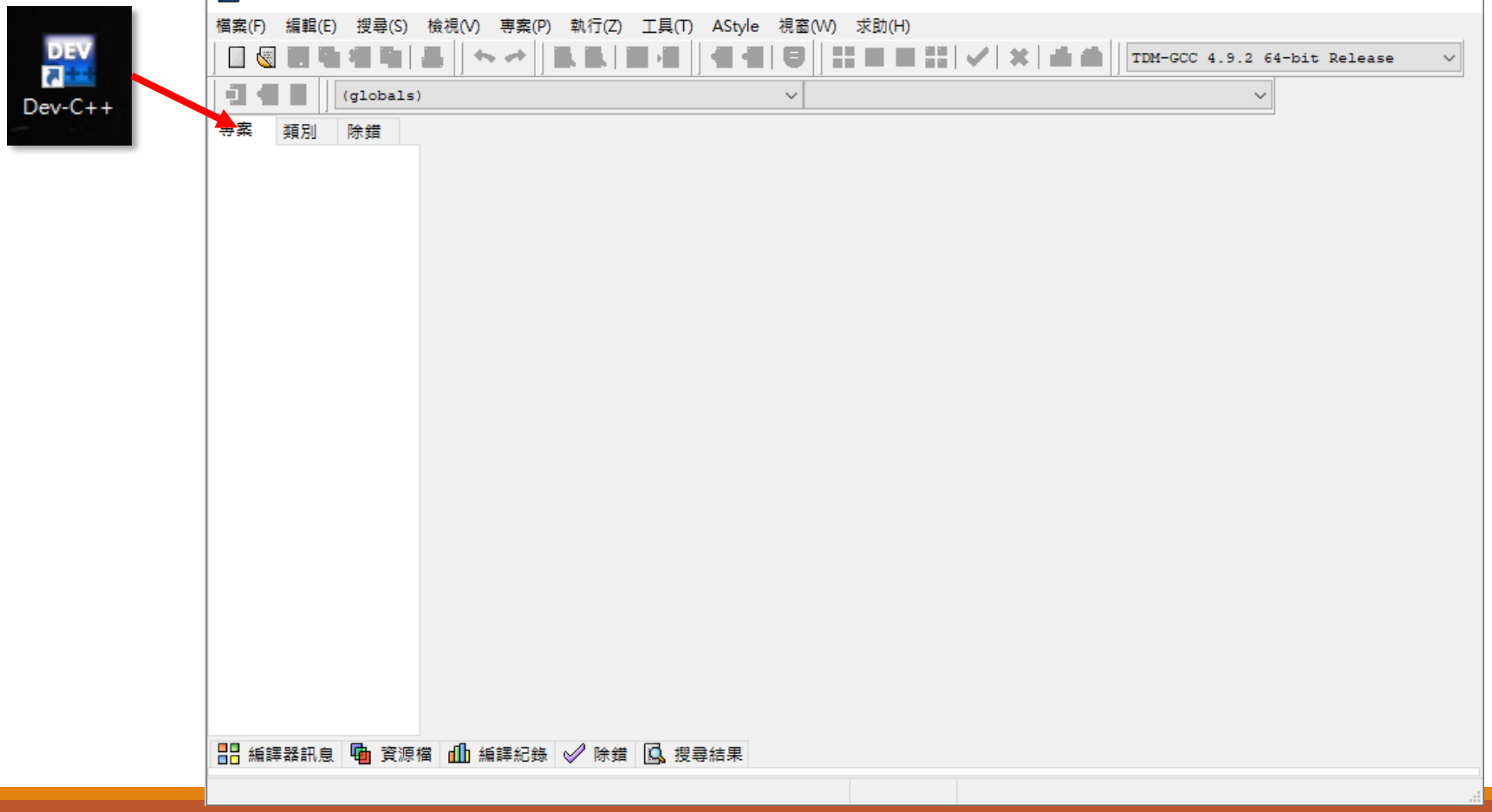
- Dev C++ 下載點
- CodeBlocks
- C++ Builder
- Microsoft Visual C++

Dev C++下載

https://sourceforge.net/projects/orwelldevcpp/



Dev C++開發環境說明


Dev-C++
Dev-C++ 5.11
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 專案(P) 執行(Z) 工具(T) AStyle 視窗(W) 求助(H)
TDM-GCC 4.9.2 64-bit Release
(globals)
專案 類別 除錯
編譯器訊息 資源檔 編譯紀錄 除錯 搜尋結果

Dev C++工具選單


功能表
編譯與執行工具列
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 專案(P) 執行(Z) 工具(T) AStyle 視窗(W) 求助(H)
(globals)
特殊工具列
主工具列
編輯工具列
搜尋工具列
專案工具列

Dev C++開啟單一程式編輯

1. 點選功能表中的「檔案／開新檔案／原始碼」選項：

2. 直接按下「主工具列」的原始碼 按鈕：

C++和VB的比較

比較	VB	C
英文字大小寫	視為相同 (實際上VB的編輯器會自行將其格式調整成一致)	視為不同
程式區塊	必要時以 End 為結尾	以大括號 {　} 包夾
每行程式結尾	以換行符號(即Enter)做標記	分號『；』做標記
註解	分號『’』後視為註解	以 // 或 /*　*/

C++程式構架

前置處理區

```
#include <iostream>
```

設定區

```
using namespace std;
```

宣告區

```
int age=15;     //公用變數宣告
void fun( );    //函式原型宣告
```

主程式區

```
int main( )
{

    return 0;  //程式結束
}
```

函式定義區

```
void fun( )     //函式實際內容
{
}
```

前置處理區

此區域為前置處理器的作業區，在這個區域中的指令會在程式被編譯前，先被前置處理器置換成某些程式碼。
最常見的是 #include 這個指令，它的作用在引入標頭檔，而所謂的標頭檔指的是包含某些函式內容的函式庫，而這些函式庫可能由編譯器本身所提供，有些是自行撰寫的函式庫。

設定區

可在此設定名稱空間std，以簡化後續程式的選寫。
名稱空間的存在是為了避免同一個程式中，使用了同樣名稱的類別或物件命名，此時就可以利用不同的名稱空間在做區隔。
例如：

std::cout

指的是在std的空間裏有一個cout的函式

宣告區

1、 **全域變數宣告**

若有變數需要提供所有的函式共同參用，就必需在此宣告。

2、 **函式宣告**

因C++的編譯器是由上到下逐行編譯，因此可在此先行宣告即將使用的函式架構，以便編譯在未編譯到函式本體時，不會把該函式視為錯誤。

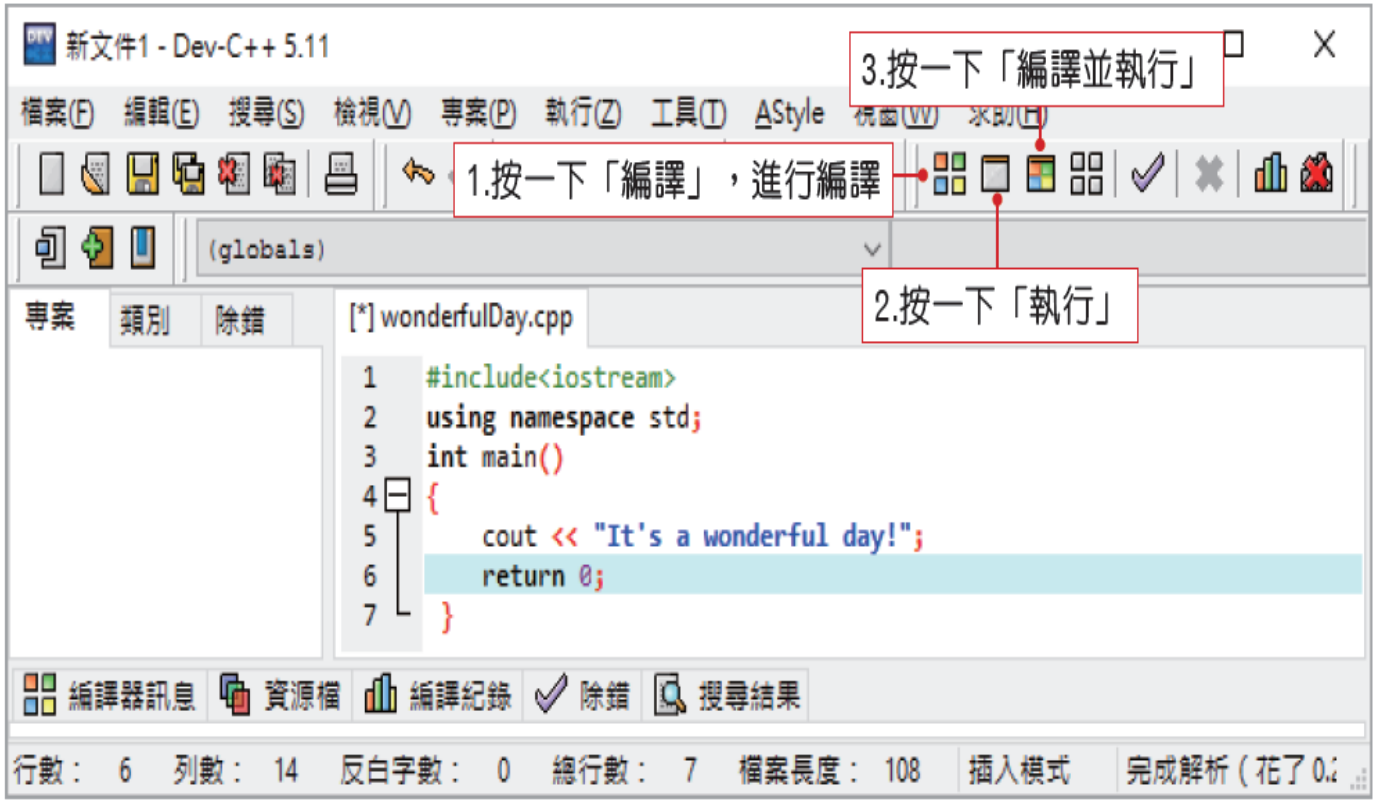
主程式區

程式的主架構，main()函式為程式的起始函式，可視為該程式和作業系統間溝通的介面。

函式定義區

為自定函式所在的區域；自定函式指的是使用者依自己的需求所編寫出來的程式段。

C++的編譯



若在編譯完馬上就要執行，就直接按3即可。

C++的程式除錯

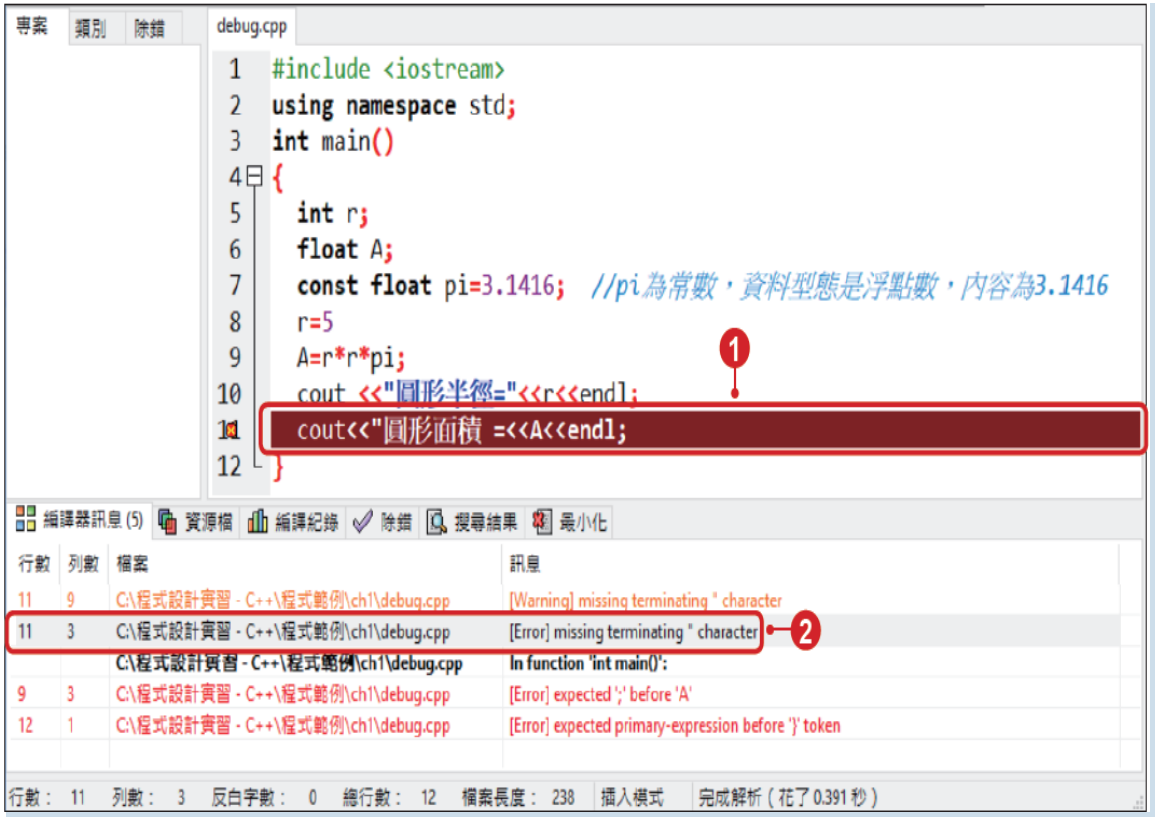
在編輯區建立這個程式。

[*] 新文件1

```
1   #include <iostream>
2   using namespace std;
3   int main()
4   {
5     int r;
6     float A;
7     const float pi=3.1416;  //pi為常數，資料型態是浮點數，內容為3.1416
8     r=5
9     A=r*r*pi;
10    cout <<"圓形半徑="<<r<<endl;
11    cout<<"圓形面積 =<<A<<endl;
12  }
```

C++的程式除錯

編譯後會產生下列錯誤結果：



錯誤說明

1、missing terminating" character

表示第11行雙引號數目不成對。

2、[Error]expect ′;′ before ′A′

表示第9 行A變數前面(即第8行)需要";"。

基本輸出函數—cout

語法－

```
cout<<變數或字串[<<……];
```

範例－

```
cout<<"X的值為："<< X;  //假設X=5
```

輸出結果－

```
X的值為：5
```

換行符號

cout輸出文字時若要做換行可在最後面加上endl或是在最後一個字串中加入控制字元\n

範例－

```
cout<<"總金額為："<< X <<"元"<<endl;
```

或

```
cout<<"X的值為："<< X <<"元\n";
```

基本輸入函數—cin

語法－

```
cin>>變數[>>變數2>> … >>變數n] ;
```

範例－

```
cin>>x;
cin>>x>>y>>z;
```

變數宣告

C++資料變數在使用前必需要先做宣告，在宣告後系統會在記憶體中配置一個空間用以儲存該筆資料。

```
void  main(){

   int   x=10;

}
```

常數和變數

- **變數－**

  為程式運算時保留資料所需之記憶體空間，其內含值會隨著程式的執行而改變，所占用的記憶體大小由最初宣告的變數型態而定。

- **常數－**

  故名思義，其值不應隨著程式之運算而改變，例如：圓周率、重力加速度等，資料一樣是存在記憶體中。

變數(常數)的資料型態

C++提供的資料型態

變數(常數)的宣告方式

- 變數宣告的語法格式－

  資料型態 變數名稱[=初值]

- 常數宣告的語法格式－

  const 資料型態 變數名稱[=初值]

  #define 常數名稱 常數值

C++變數及常數的命名通則

識別字	命名原則	範例
變數	第一個字元為小寫英文字母 若由數個英文單字組成，則後面英文單字的第一個字母為大寫，其餘則為小寫	ratio applesNum
常數	全部字元皆為英文大寫字母與底線	PI PASS_SCORE

整數型態

中文名稱	資料型態	位元組	表示範圍
短整數	short int	2	-32,768～32,767
	unsigned short int	2	0～65,535
整數	int	4	-2,147,483,648～2,147,483,647
	unsigned int	4	0～4,294,967,295
長整數	long int	4	-2,147,483,648～2,147,483,647
	unsigned long int	4	0～4,294,967,295

浮點數型態

資料型態	中文名稱	位元組	有效位數	表示範圍
float	浮點數	4	7	正數： $1.401298 \times 10^{-45} \sim 3.402823 \times 10^{38}$ 負數： $-3.402823 \times 10^{38} \sim -1.401298 \times 10^{-45}$
double	倍精確度浮點數	8	15	正數： $4.94065645841247 \times 10^{-324}$ $\sim 1.79769313486232 \times 10^{308}$ 負數： $-1.79769313486232 \times 10^{308}$ $\sim -4.94065645841247 \times 10^{-324}$

浮點數的表示

當浮點數大或小到一個程度時，系統在輸出資料時會以科學記號的方式表示：

$m \times 10^{\pm n}$（數學表示格式）

mE±n（電腦程式表示格式）

m為假數(mantissa)

n為指數，代表10的n次方

數字	科學記號	在C++中的表示法
5,000,000	5×10^{6}	5E6(或5e6)
-369,000	-3.69×10^{5}	-3.69E5
123.45	1.2345×10^{2}	1.2345E2
0.0000078	7.8×10^{-6}	7.8E-6
-0.00123	-1.23×10^{-3}	-1.23E-3

字元資料(ASCII)型態

1、 宣告時所用的字元為 char。

2、 字元資料型態的長度為 1byte，編碼的範圍是0~255。

3、 0到127的編碼範圍稱為「內碼」，是比較通用的符號或控制碼，例如：大寫A的ASCII碼是65，小寫a則是97。

4、 編碼128到255的部分為「延伸ASCII碼」(extended ASCII)，包括數學符號與表格框線……等。

布林資料型態

1、 宣告時所用的字元為 bool。

2、 布林(boolean)型態的變數，通常用在程式的流程控制，其資料值只可以是1(true，真)或0(false，假)兩種之一。

3、 若要宣告布林變數condition，並設定初始值為1，可以寫出如下的敘述：

```
bool condition=1;
bool condition=true;
```

變數宣告範例

```
int x;
int x=100;
int x=0,y=0,z;
float k=0;
char x;
char s=“abc”;
Bool t=true;
const float PI=3.14;
```

C++的資料型態轉換

C++有兩種轉換資料型態的方法：

1、自動型態轉換

2、強制型態轉換

自動型態轉換

1、在執行指定(‘=’)運算的時候，若等號兩邊的變數資料型態不同，編譯器會自動轉換變數的資料型態。

2、原則是先將等號右邊變數的資料型態轉換成等號左邊變數的資料型態，再執行指定的動作。

3、此類型的轉換亦稱隱含型態轉換(implicit type conversion)。

自動型態轉換說明

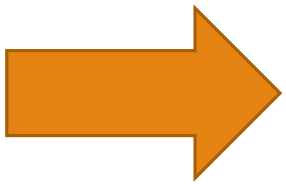
```
void  main(){

   int   x;
   int   k=3.7;

   x=k;

  return 0;

}
```



執行後

此範例是要將浮點數存入整數空間，依自動轉換原則，k的值在存入x前系統會將先變小數去除後(注意，k的內容並不會改變)，存入x，最後x=3

強制型態轉換(一)

當算術運算式中變數的資料型態不同時，C++編譯器會自動將可表示範圍較小的資料型態轉換成可表示範圍較大的資料型態，再進行運算

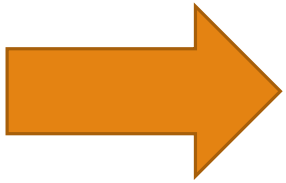
```
void  main(){

   int   x;
   int   k=3.7;

   k=k+x;

  return 0;

}
```



執行後

浮點型態和整數型態相加，因浮點型態較大，所以系統將x轉成浮點型態後再和k相加。

強制型態轉換(二)

在C++中，兩筆整數相除所得到的結果為商(整數)，小數的部分則會被省略。

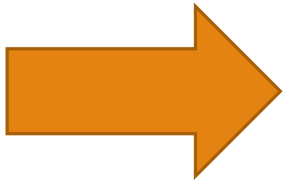
```
void  main(){

   int   x;
   int   k=3.7;

   k=7/3;

  return 0;

}
```



執行後

雖然除完後存入浮點型態的變數，但因除式中兩數皆為整數，因此系統只取其商並將其存入k中，執行後k=2。

強制型態轉換(三)

若兩整數相除時，所得之結果想要保留小數，則必需在除之前先將其中一個數先轉為浮點型態。

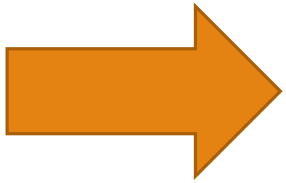
```
void  main(){

   int   x;
   int   k=3.7;

   k=(float)7/3;

  return 0;

}
```

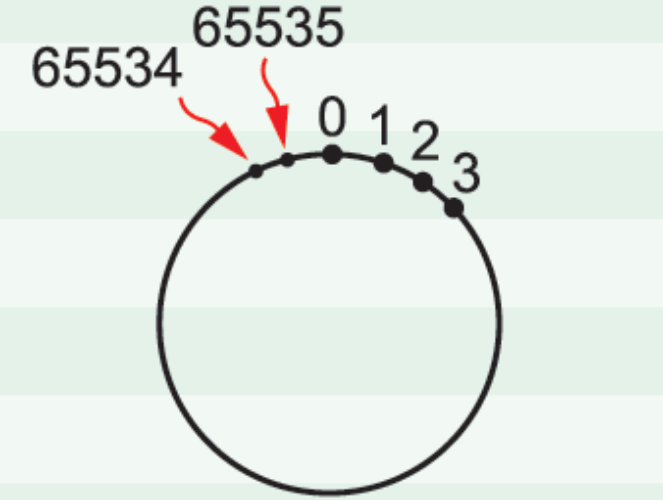


執行後

式子中7已被轉成浮點型態，因此，系統在做除法時會將浮點型態來做除法，因此結果的小數會被保留，執行後k=2.3333。

C++的溢位處理

當兩數運算時，若超出其資料型態的表達範圍，在VB中會直接停止程式執行並出現警告，而在C++中仍會執行運算，但其結果是錯誤的，其運算原則如下圖所示：



unsigned short 資料型態的表示範圍為0~65535。所以加1表示依順時針方向移一格；減1表示依逆時針方向移一格

C++的運算式

- 算數運算
- 遞增、遞減運算
- 關係運算
- 邏輯運算
- 位元運算
- 移位運算
- 複合運算

算數運算子

運算子	意義	範例	說明
()	括號	a=(t+b)*h/2	括號中的運算子會優先執行，本例為計算梯形的面積，若忽略括號則會依先乘除後加減的規則來運算。
*	乘法	total=5*8	先計算5*8(結果為40)，再將total指定為40。
/	除法	avg=3/2	使用除號應注意資料型態的轉換，設計程式時，請先思考avg應該宣告成浮點數或宣告成整數。 avg=3/2 ⇨ avg=1(整數除以整數，結果為商數) avg=3.0/2 ⇨ avg=1.5(浮點數除以整數) avg=3/2.0 ⇨ avg=1.5(整數除以浮點數) avg=3.0/2.0 ⇨ avg=1.5(浮點數除以浮點數)
%	取餘數	mod=7 % 2	先計算7÷2的餘數後，再把結果指定給mod儲存。
+	加法	sum=sum+i	先將sum加i後，再把結果指定給sum儲存。
-	減法	d=a-b	先將a減b後，再把結果指定給d儲存。

遞增、遞減運算子

運算子	意義	範例	說明
++	遞增變數值(變數值+1)	i++	相當於i=i+1，即先將i加1後，再把結果指定給i儲存
--	遞減變數值(變數值-1)	i--	相當於i=i-1，即先將i減1後，再把結果指定給i儲存

註：

若於多算式中++放在變數前和變數後會有執行順序的問題，如(假設i的值為2)：

```
x=i++*5;
```

x=10、i=3

```
x=++i*5;
```

x=15、i=3

遞增、遞減運算子說明

敘述	等效敘述 (拆解成兩個步驟)	執行後的變數值
a=i++	①a=i ②i=i+1	①a=5 ②i=6
a=++i	①i=i+1 ②a=i	①i=6 ②a=6
a=i--	①a=i ②i=i-1	①a=5 ②i=4
a=--i	①i=i-1 ②a=i	①i=4 ②a=4

關係運算

運算子	意義	範例	說明
>	大於	5>3	5>3⇨5大於3，所以得到結果為1(true)
>=	大於等於	4>=5	4>=5⇨5並不大於等於5，所以得到結果為0(false)
<	小於	6<8	6<8⇨6小於8，所以得到結果為1(true)
<=	小於等於	7<=6	7<=6⇨7並不小於等於6，所以得到結果為0(false)
==	等於	5==5	5==5⇨5等於5，所以得到結果為1(true)
!=	不等於	7!=4	7!=4⇨7不等於4，所以得到結果為1(true)

邏輯運算

邏輯運算的結果非真(true)即假(false)，主要用於判斷指令中

運算子	意義	範例	說明
&&	AND，及，且	x && y	x AND y
||	OR，或	x || y	x OR y
!	NOT，反	!x	NOT x

位元運算

位元運算是將數值化為二進制的型態後，再逐位元去做布林運算。

位元邏輯運算子	意義
~	位元NOT運算子
&	位元AND運算子
|	位元OR運算子
^	位元XOR運算子

移位運算

運算子	意義	範例	說明
<<	左移	10<<2	將10往左移2位元，結果為10*4=40
>>	右移	10>>1	將10往右移1位元，結果為10/2=5

1、 左移是將數值的化成二進位型態後將位元向左移動n個位元。

2、 向左移動後，超出儲存範圍的數字捨去，右邊空白位元則補0。

複合運算

運算子	意義	範例	等效語法
+=	加法指定運算	A+=3	A=A+3
-=	減法指定運算	A-=3	A=A-3
*=	乘法指定運算	A*=3	A=A*3
/=	除法指定運算	A/=3	A=A/3
%=	餘數指定運算	A%=3	A=A%3
^=	XOR位元指定運算	A^=B	A=A^B
&=	AND位元指定運算	A&=B	A=A&B
|=	OR位元指定運算	A|=B	A=A|B
<<=	位元左移指定運算	A<<=2	A=A<<2
>>=	位元右移指定運算	A>>=2	A=A>>2

運算式的優先順序

優先順序	運算子	結合性	運算類別
1	()	由左至右	括號運算子
2	!、+、-	由右至左	一元運算子
	~	由右至左	位元邏輯運算子
	++、--	由右至左	遞增遞減運算子
3	*、/、%	由左至右	算術運算子
4	+、-	由左至右	算術運算子
5	<<、>>	由左至右	位元左移右移運算子
6	>、>=、<、<=	由左至右	關係運算子
7	==、!=	由左至右	關係運算子
8	&	由左至右	位元邏輯運算子
9	^	由左至右	位元邏輯運算子
10	|	由左至右	位元邏輯運算子
11	&&	由左至右	邏輯運算子
12	||	由左至右	邏輯運算子
13	?:	由右至左	條件運算子
14	=	由右至左	指定運算子