

C++ and Quantitative Finance

交通大學
資訊與財金管理學系
戴天時
cameldai@mail.NCTU.edu.tw

1

課程簡介

時間	主題	實務演練
	C++語言概論	現值的計算 商業本票價格計算
	程式流程控制及迴圈	債券價格和存續期間計算 使用二分法計算 IRR Yield spread 的計算
	陣列結構	計算零息利率, 遠期利率, static spread, 矩陣計算, 投資組合的風險值估計
	函式的撰寫及呼叫	Black-Scholes formula, 隱含波動度, 新奇選擇權評價

2

課程簡介

主題	實務演練
編譯器介紹, 字串及檔案的處理	驗證葛蘭碧移動平均線法則 股票價格歷史波動度計算
樹狀結構評價, 修改以及組合數學	二元樹理論, 評價程式撰寫, 美式選擇權評價, 使用組合數學加速評價, 新奇選擇權的評價
蒙地卡羅法	使用蒙地卡羅法評價選擇權, 縮減評價誤差的技巧, 彩虹選擇權評價
物件導向程式設計	障礙選擇權的程式碼再利用
DLL的建構	使用Excel呼叫DLL

3

課程資源

- 範例程式: 請參見硬碟的資料夾
- 參考書目或講義:
 - C++財務程式設計 戴天時 (2005)
證券暨期貨發展基金會
ISBN 986-7737-51-2

4

Acknowledgement

- 感謝下列人士為改寫本教材所投入的心力

台大資訊所	許智睿	AMM
台大財金所	陳宜廷	Quasi-Monte Carlo simulation
台大資訊所	章宇傑	C++ DLL的建立以及如何使用Excel呼叫DLL,XLL的撰寫
台大資訊所	邱俊淵	
台大資訊所	王鈞茹	

5

C++語言概論

C++語言簡介
編譯程式簡介
撰寫你的第一個C++程式
資料型態和運算式
程式檢驗和除錯
實務演練:現值運算,商業本票價值計算
算術運算式的簡寫
邏輯運算式

6

C++ 語言歷史簡介

- C
 - 從BCPL 和 B這兩個語言演變而來
 - 用來開發作業系統
 - 最重要的程式語言
- C++
 - 由Bell Labs 的Bjarne Stroustrup開發
 - 包含C語言的架構
 - 加入物件導向的概念→程式開發更快速
 - 先學會C再進入C++

7

C++語言和BASIC和 JAVA比較

C++	Basic	JAVA
<ul style="list-style-type: none">•執行效率高•功能強大•透過CGI支援動態網頁功能	<ul style="list-style-type: none">•易入手•Excel 提供VBA(執行效率慢)	<ul style="list-style-type: none">•執行較慢•易於撰寫網頁及網路程式•可有效降低網頁伺服器負擔

8

常見程式語言之比較

	C/C++	JAVA	Visual BASIC	C# / .NET	Matlab
優點	<ul style="list-style-type: none"> 執行效率高 功能強大 	<ul style="list-style-type: none"> 跨平台 Sun提供大量函式庫 	<ul style="list-style-type: none"> 易學 易於開發視窗程式 	<ul style="list-style-type: none"> 跨平台 微軟提供大量函式庫 易於開發視窗程式 	<ul style="list-style-type: none"> 專為科學計算設計的語言 龐大的科學計算用函式庫
缺點	<ul style="list-style-type: none"> 不易精通 	<ul style="list-style-type: none"> 執行較慢 	<ul style="list-style-type: none"> 執行較慢 	<ul style="list-style-type: none"> 執行較慢 	<ul style="list-style-type: none"> 執行較慢
網頁	<ul style="list-style-type: none"> 透過CGI 	<ul style="list-style-type: none"> JSP 	<ul style="list-style-type: none"> ASP 	<ul style="list-style-type: none"> ASP.NET 	<ul style="list-style-type: none"> 無
特殊能力	<ul style="list-style-type: none"> 撰寫作業系統 	<ul style="list-style-type: none"> Java Applet 跨平台網頁程式，減低伺服器負擔 	<ul style="list-style-type: none"> Excel 附帶VBA，可開發Excel 程式 	<ul style="list-style-type: none"> 可支援 VB 語法，稱為VB.NET 	<ul style="list-style-type: none"> 矩陣運算 繪製函數

C/C++

- C++是一種使用非常廣泛的電腦程式設計語言。C++是一種靜態資料類型檢查的，支援多範型的通用程式設計語言。它支援程式化程式設計、資料抽象化、物件導向程式設計、泛型程式設計、基於原則設計等多種程式設計風格。
- 一開始C++是作為C語言的增強版出現的，從給C語言增加類別開始，不斷的增加新特性。virtual function、operator overloading、multiple inheritance、template、例外exception、RTTI、namespace逐漸納入標準。
- 根據《Thinking in C++》一書所評述的，C++與C的效率往往相差在正負5%之間。所以有人認為在大多數場合中，C++完全可以取代C語言。

JAVA



- Java 編程語言的風格十分接近C++語言。繼承了 C++ 語言面向對象技術的核心，Java捨棄了C++語言中容易引起錯誤的指標、operator overloading、多重繼承等特性，增加了垃圾回收器功能用於回收不再被引用的對象所佔據的內存空間。
- Java首先將源代碼編譯成bytecode，然後依賴各種不同平台上的虛擬機來解釋執行字節碼，從而實現了「一次編譯、到處執行」的跨平台特性。不過，這同時也在一定程度上降低了 Java 程序的運行效率。
- Sun公司對Java編程語言的解釋是：Java編程語言是個簡單、面向對象、分佈式、解釋性、健壯、安全與系統無關、可移植、高性能、多線程和動態的語言。

BASIC

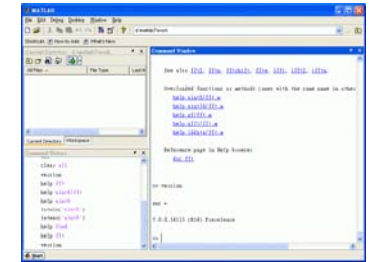
- BASIC (Beginner's All-purpose Symbolic Instruction Code)，意思就是「初學者的全方位符式指令代碼」，是一種設計給初學者使用的程式語言。BASIC是一種解譯式的程式語言，在完成編寫後不須經由編譯及連結等手續即可執行，但如果需要單獨執行時仍然需要將其建立成執行檔。
- 1991年，伴隨著MS-DOS5.0的推出，微軟公司同時推出了Quick BASIC的簡化版QBASIC，將其作為作業系統的組成部分免費提供給用戶。自從Windows作業系統出現以來，圖形用戶界面 (GUI) 的BASIC語言 (即Visual Basic) 已經得到廣泛應用。

C# / .NET

- **C#**（發音為C Sharp）是由微軟公司所開發的一種物件導向，且運作于.NET Framework之上的高階程式語言。並且成為ECMA與ISO標準規範。**C#**看似基於C++寫成，但又融入其它語言如Delphi、Java、VB等。
- Wikipedia 介紹：
 - http://zh.wikipedia.org/w/index.php?title=C_Sharp&variant=zh-tw

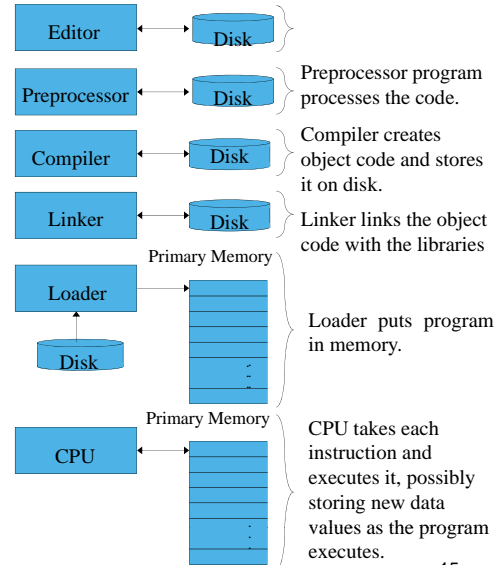
MATLAB

- **MATLAB**基於矩陣（Matrix）運算，其全稱MATrix LABoratory。它在數學類科技應用軟體中在數值計算方面首屈一指。
- 目前 **MATLAB** 產品族可以用來進行以下各種工作：
 - 數值分析
 - 數值和符號計算
 - 工程與科學繪圖
 - 控制系統的設計與模擬
 - 數字圖像處理
 - 數字信號處理
 - 通訊系統設計與模擬
 - 財務與金融工程



C++程式開發流程

- 1. 編輯程式
- 2. 前置處理 (Preprocess)
- 3. 編譯
- 4. 聯結函式庫
- 5. 呼叫程式執行



編譯程式介紹

- 使用程式: Visual Studio 2015
- 本課程著重在使用C++程式撰寫財務數值程式

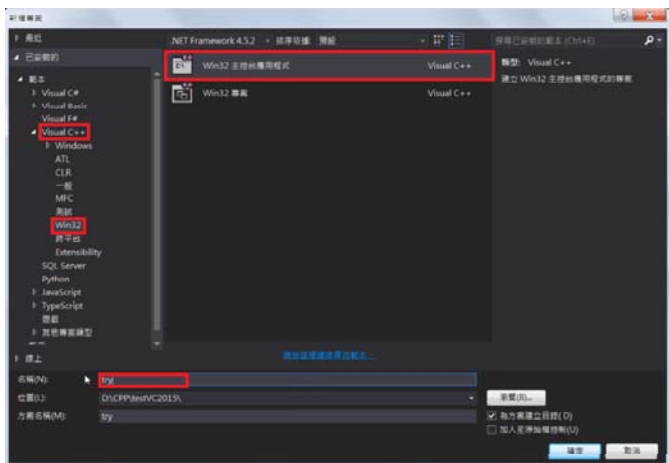
使用Visual C++撰寫第一個程式 (建立一個新的Project)

- 開啓一個新的 project
 - 檔案->新增 ->專案->選擇“Win32 主控台應用程式”並輸入專案名稱 ->確定 ->按“下一步” ->勾選“空專案” ->按“完成”

參照圖例



參照圖例



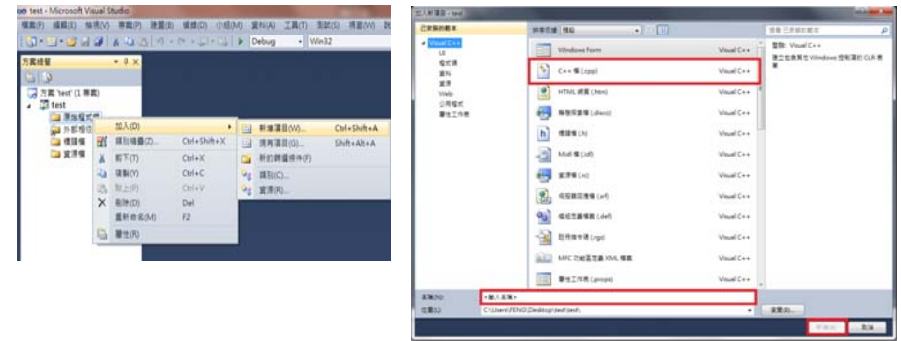
參照圖例



使用Visual C++撰寫第一個程式 (編譯並執行你的程式)

- 新增一個新的 **cpp**檔
 - 原始程式檔按右鍵 ->加入 ->新增項目->選擇“C++檔(.cpp)”並輸入名稱 ->按新增
 - 將程式碼打完後 ->按“F7(建置方案)”開始編譯
 - 編譯成功後，再按“F5(開始偵錯)”或者“Ctrl+F5(啟動但不偵錯)”即可執行程式

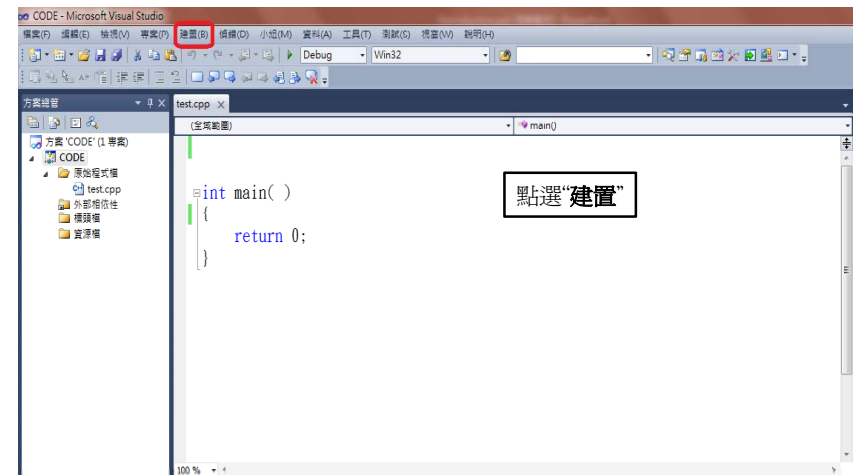
參照圖例



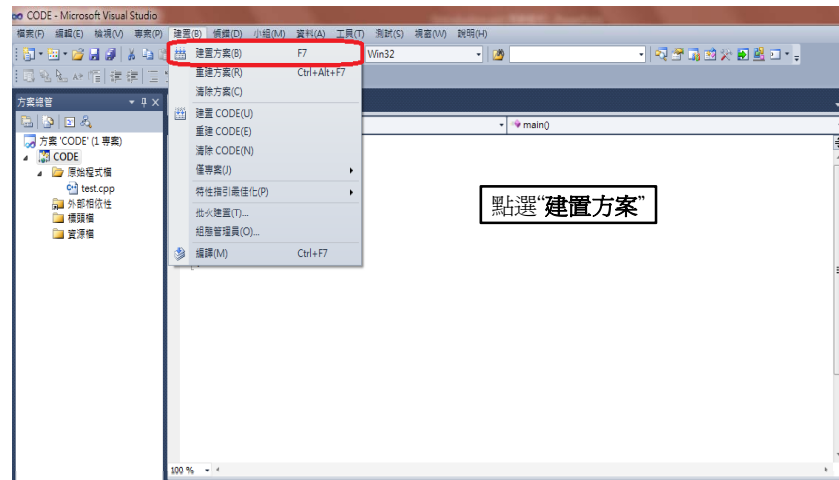
主函式 main function

- 主函式是程式開始執行的地方。
- `int main()`
`{`
`return 0;`
`}`
- `int`表示主函式回傳整數型態，`()`內放入參數，`{}`表示函式的作用範圍。

程式編譯



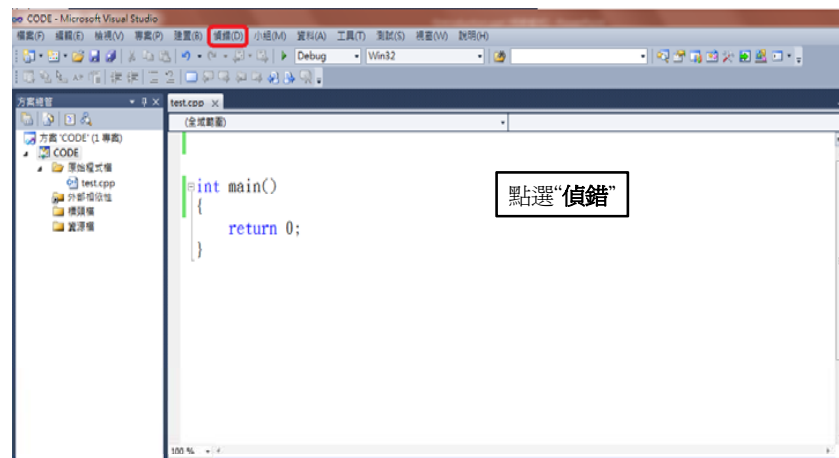
程式編譯



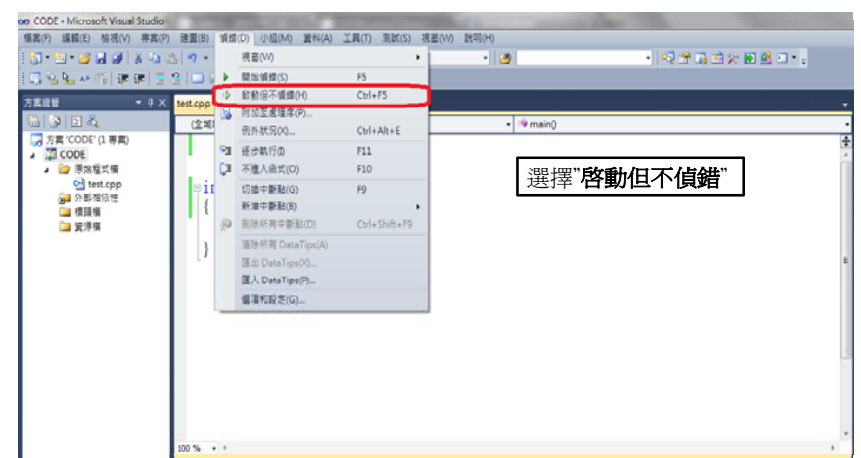
程式編譯



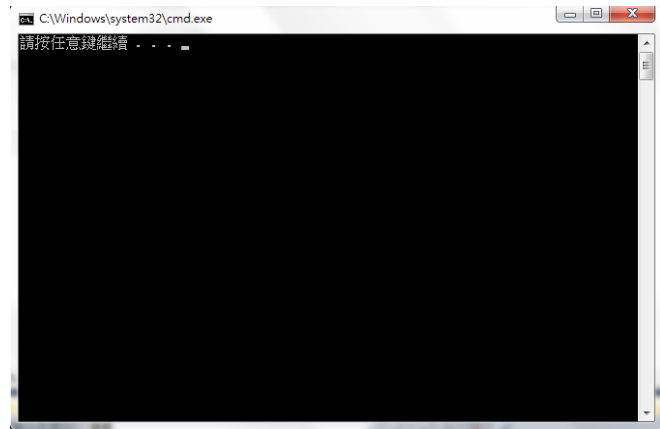
程式執行



程式執行



執行結果



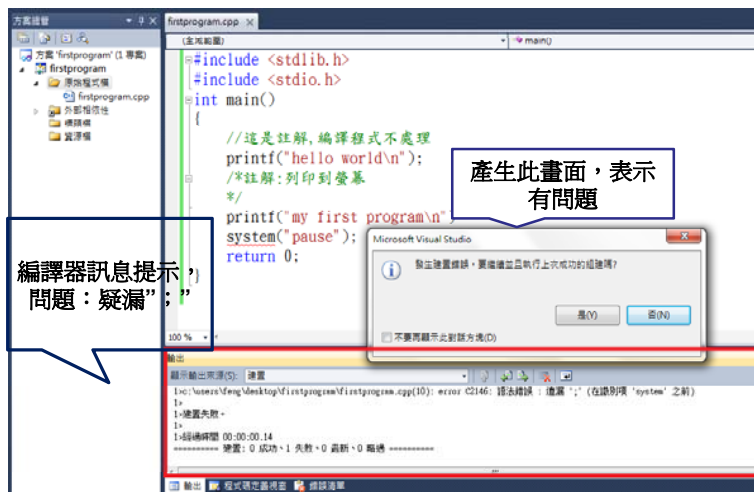
編輯程式 見 FirstProgram project

```
#include <stdlib.h> → 引用函式庫:stdlib(呼叫system)
#include <stdio.h> → 引用函式庫:stdio(呼叫printf)

int main()
{
    //這是註解,編譯程式不處理
    printf("hello world\n"); → \n代表換行
    /*註解:列印到螢幕          列印字串:使用printf
    */
    printf("my first program\n"); ← 用分號:來分開指令
    system("pause"); ←
    return 0; ← 程式執行到這個指令,可將程式畫面暫停,按下任意鍵再繼續
}
    ↑
    程式執行結束時會傳回一個整數值
```

不同顏色代表不同文字屬性

程式產生錯誤訊息的畫面



常見錯誤訊息

錯誤	顯示訊息
include拼錯	error C1021: 無效的前置處理器命令 'include' ('include'代表拼錯的名字) IntelliSense: 無法辨認的前置處理器指示詞
函式庫(stdio)名字拼錯	IntelliSense: 無法開啓 來源 檔案 "studio.h" ('studio.h': 代表拼錯的名字)
主函式名字(main)拼錯	error LNK2019: 無法解析的外部符號 _main 在函式 __tmainCRTStartup 中被參考
分號漏打	IntelliSense: 必須是 ';'
printf函式名稱拼錯	IntelliSense: 識別項 "prntf" 未定義 ('prntf'代表拼錯的名字)
漏打右大括弧 }	error C1075: 與左大括號 '{' 對應之前遇到檔案的結尾

課堂演練

- 開啓一個新的project
- 使用printf來列印出自己的工作單位和名稱
例: `printf("交大資財\n");`
`printf("戴天時\n");`

33

變數的資料型態

- 變數可視為一塊可儲存資料的記憶體空間
- 變數包含了變數的名稱和資料型態
- 常見資料型態

型態	char	int	float	double
意義	字元	整數	浮點數	倍精準浮點數

- 變數需經宣告才可使用:

`int a,b,c;` 宣告三個整數 `a,b,c`

34

變數的命名

- 由任意連續的字母、數字或是底線組成
 - 其他符號如@、#、\$......等，都不可出現在變數名稱
 - `var#7` is invalid
 - 第一個字不得為數字
 - `1Var` is invalid
 - 保留字不可重複使用

C++ Reserved Words							
asm	auto	break	case	catch	char	class	const
continue	default	delete	do	double	else	enum	extern
float	for	friend	goto	if	inline	int	long
signed	sizeof	static	struct	switch	template	this	throw
try	typedef	union	unsigned	virtual	void	volatile	while

不可宣告
`int double;`

35

變數的命名

- 大小寫區分
 - `Abc` 和 `abc` 是不同變數
- 變數命名需有意義
 - `FirmValue=DebtValue+EquityValue`
 - 不用 `a=b+c` // 程式可讀性
- 課堂練習: 說明下列命名不合法理由
 - `1AB`
 - `A!B`
 - `float`

36

資料的輸入輸出

- 輸入:使用scanf, 輸出: 使用 printf

型態	char	int	float	double
符號	%c	%d	%f	%lf

見 Variable project

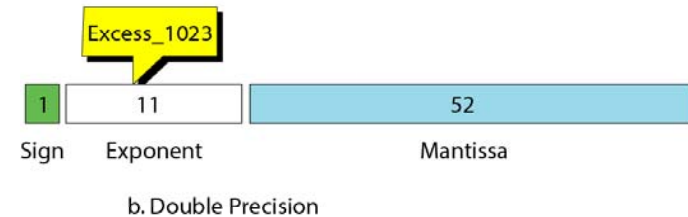
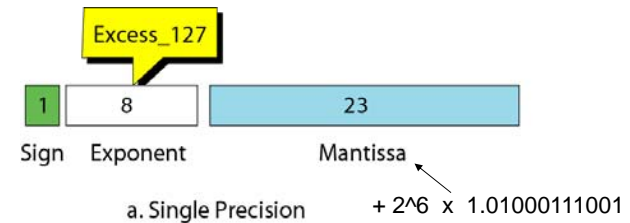
- 輸入兩個整數,輸出這兩個數的和

```
int a,b;          宣告兩個整數 a,b
char c;          宣告變數 c
scanf("%d",&a);  輸入 a "%d"代表整數
printf("a=%d\n",a); 結果輸出
printf("b=%d\n",b);  b的值尚未初始化,為一亂數
```

Warning message

37

IEEE standards for floating-point representation



38

課堂演練

- 宣告一倍精密度浮點數 xx.
 - double xx;
- 用 scanf() 輸入 xx
 - 比較

```
scanf("%lf", &xx);
printf("%lf",xx);
```

和

```
scanf("%f", &xx);
printf("%lf",xx);
```

的輸出結果

39

資料型態和變數值的關係

- ```
b=2.4;
printf("b=%d",b); b的值為2 (小數點後捨去)
c=300;
printf("c=%d",c); c的值不為300 (超出儲存空間)
```

| 資料型態         | 大小(位元組) | 簡述                                       |
|--------------|---------|------------------------------------------|
| char         | 1       | 表示字元                                     |
| short        | 2       | short代表short integer, 表示用所占空間大小為16bit的整數 |
| int (或 long) | 4       | long或是int皆表示用所占空間大小為32bit的整數             |
| float        | 4       | float表示用所占空間大小為32bit的浮點數                 |
| double       | 8       | double表示用所占空間大小為64bit的浮點數                |

$2^8=256 \Rightarrow -128\sim 127$

40

## 算術與邏輯運算式

- C++允許算術運算以及邏輯運算式出現在同一個算式中

### – 算術運算

- 加(+), 減(-), 乘(\*), 除(/), 取餘數 (%)
- `double d,e; d = 1+a*b%2;` = 為指定運算元, 將 `1+a*b%2` 的值放入 `d`
- `d=1+a*b%2;` 先算 `a*b`, 再算 `(a*b)%2`, 最後計算
- 運算可分成浮點數和整數的計算
- `e=2/3;`
- `e=2.0/3;` 整數除法
- `e=a/b;`
- `e=(double) a/b;` 型態強制轉換, 可寫成 `e=double(a)/b;`

Don't write `3=1+2`

41

## 課堂演練 程式碼比較

- Case 1:
  - `double a=2.4;`
  - `double b=1/2*a;`
  - `b=?`
- Case 2:
  - `double a=2.4;`
  - `double b=a*1/2;`
  - `b=?`

42

## 折現值計算

- 列出運算式: (折現值的計算)
  - 假定利率 `r%`, 考慮一年後支付 `d` 元, 兩年後支付 `e` 元的現金流量的現值

- 程式範例如下:

```
int d,e,f; 見PV project
float r;
scanf("%d",&d); 輸入現金流 d,e 和利率 r
scanf("%d",&e);
scanf("%f",&r);
f=d/(1+r); 計算d的現值
f=f+e/((1+r)*(1+r)); 加上e的現值
printf("Present value=%f",f); 列印結果
```

- 程式語法正確, 但是計算結果不正確
  - 程式除錯(debug)

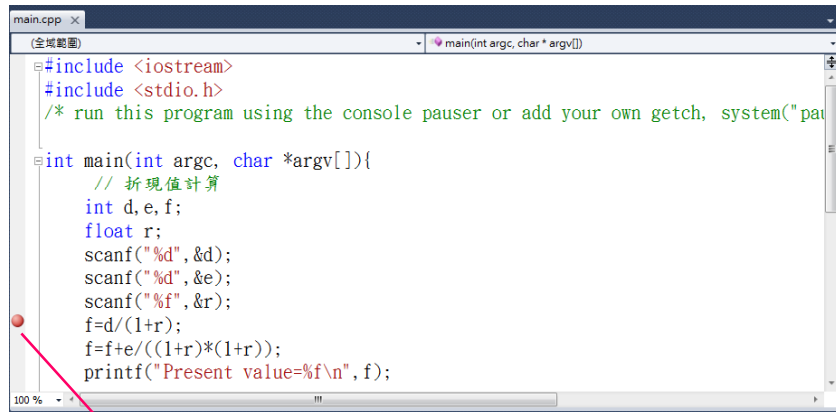
43

## 程式單步執行檢驗和除錯

- 首先, 在程式的進入點設定 **Breaking Point**
    - 在行號上點一下左鍵即可, 會出現一個紅點
  - 再來, 點選上方功能表“偵錯”
    - 選“開始偵錯”, 或者直接按“F5”
    - 利用偵錯功能表的“不進入函式”, 或者直接按“F10”來逐步偵錯
- ```
f=d/(1+r);
f=f+e/((1+r)*(1+r));
```
- `f` 宣告為整數, 無法儲存小數
 - 修改 `f` 的資料型態
- 重新執行程式

44

程式除錯流程

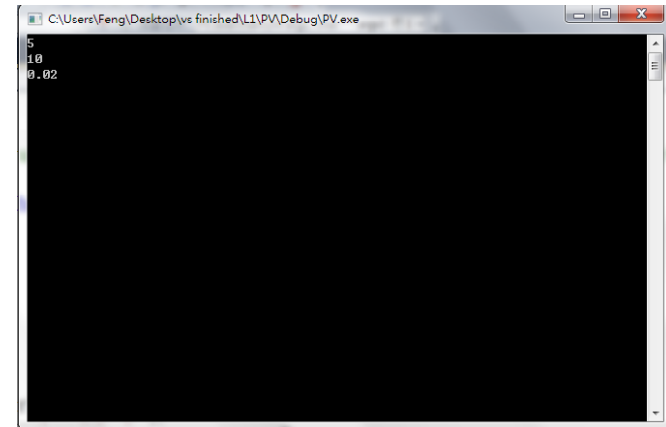


```
#include <iostream>
#include <stdio.h>
/* run this program using the console pauser or add your own getch, system("pause") or input loop */

int main(int argc, char *argv[]){
    // 折現值計算
    int d, e, f;
    float r;
    scanf("%d", &d);
    scanf("%d", &e);
    scanf("%f", &r);
    f=d/(1+r);
    f=f+e/((1+r)*(1+r));
    printf("Present value=%f\n", f);
}
```

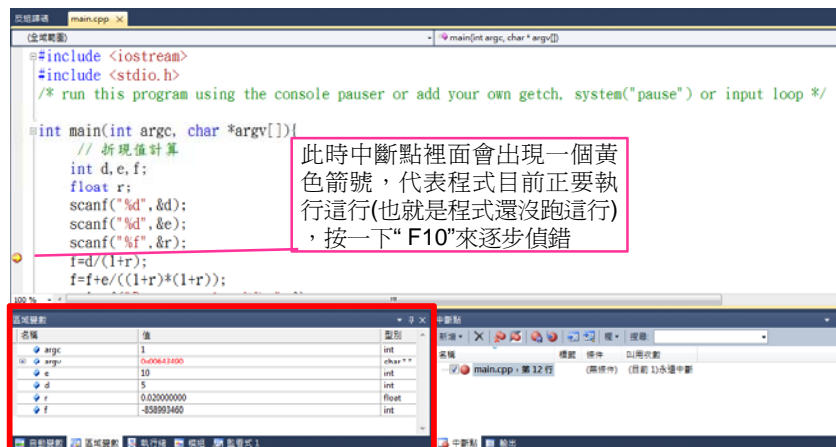
在此行按一下滑鼠左鍵，即可設定中斷點。並按“F5”開始偵錯。

程式除錯流程



按“F5”開始偵錯後，會要求輸入變數d、e、r的值。這邊舉例d=5、e=10、r=0.02

程式除錯流程



```
#include <iostream>
#include <stdio.h>
/* run this program using the console pauser or add your own getch, system("pause") or input loop */

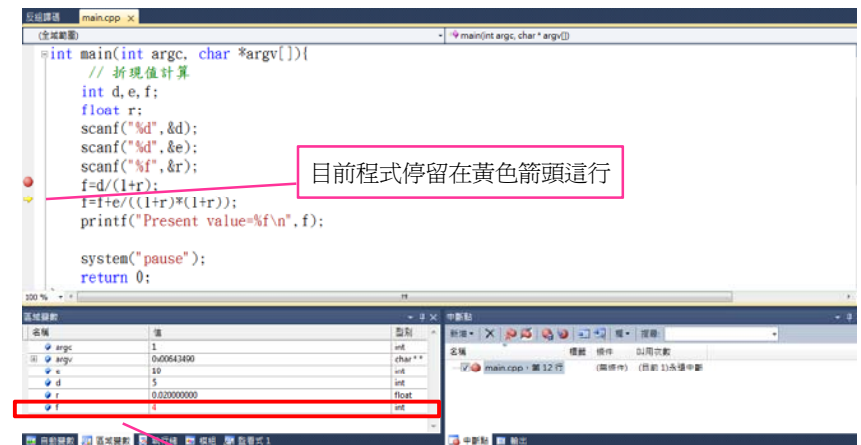
int main(int argc, char *argv[]){
    // 折現值計算
    int d, e, f;
    float r;
    scanf("%d", &d);
    scanf("%d", &e);
    scanf("%f", &r);
    f=d/(1+r);
    f=f+e/((1+r)*(1+r));
}
```

此時中斷點裡面會出現一個黃色箭號，代表程式目前正在執行這行(也就是程式還沒跑這行)，按一下“F10”來逐步偵錯

名稱	值	型別
argc	1	int
argv	0x0041400	char**
e	10	int
d	5	int
r	0.02000000	float
f	-458983460	int

此時變數d、e、r已有初始值，因為尚未執行f=d/(1+r)，f的值為一奇怪的數字

程式除錯流程



```
int main(int argc, char *argv[]){
    // 折現值計算
    int d, e, f;
    float r;
    scanf("%d", &d);
    scanf("%d", &e);
    scanf("%f", &r);
    f=d/(1+r);
    f=f+e/((1+r)*(1+r));
    printf("Present value=%f\n", f);

    system("pause");
    return 0;
}
```

目前程式停留在黃色箭頭這行

名稱	值	型別
argc	1	int
argv	0x00643400	char**
e	10	int
d	5	int
r	0.02000000	float
f	4	int

執行完f=d/(1+r)，f的值為4，因為f的型態是整數，所以將浮點數運算結果存為整數，因此需將f型態改為浮點數即可

程式除錯流程



課堂演練:程式除錯 商業本票價值計算

- 由企業所發行無擔保的短期債券
- 根據企業在市場上的信用等級以及當時的市場利率折價發行

$$\text{面額十萬元商業本票價格} = 100000 - (100000 \times \text{發行利率} \times \frac{\text{發行天期}}{365})$$

- CommericalPaper project程式
 - 輸入天數,發行利率
 - 計算本票價格
- 程式結果不正確 → 使用上述除錯工具除錯

算術與邏輯運算式 算式的簡寫

請參見Variable project Part 4

- $a+=b \rightarrow a=a+b;$
 - ◆ $a=b; a*=b; a/=b; \rightarrow a=a-b; a=a*b; a=a/b;$
- $a+=(b-a); \rightarrow a=a+b-a; \rightarrow a=b;$
- $a++ \rightarrow a=a+1;$
- $a-- \rightarrow a=a-1;$

算術與邏輯運算式

- 邏輯運算(常用來處理程式的流程控制)
- non zero → TRUE(1), zero → FALSE(0)

NOT (!)		AND (&&)		
x	NOTx	x	y	x AND y
0	1	0	0	0
1	0	0	1	0
		1	0	0
		1	1	1

OR ()		XOR			
x	y	x OR y	x	y	x XOR y
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

算術與邏輯運算式

- 邏輯算式的撰寫

```
double rate1=0.011, rate2=0.007;
```

```
(rate1<0.01)&&(rate1>0.005)
```

- Range notes

```
(rate1<0.01)||rate2<0.01)
```

- Barrier option

53

算術與邏輯運算式

- 計算的優先順序

Operators						Associativity	Type
++	--	+	-	!	(type)	right to left	unary
*	/	%				left to right	multiplicative
+	-					left to right	additive
<	<=	>	>=			left to right	relational
==	!=					left to right	equality
&&						left to right	logical AND
						left to right	logical OR
?:						right to left	conditional
=	+=	-=	*=	/=	%=	right to left	assignment
,						left to right	comma

Fig. 4.16 Operator precedence and associativity.

Ex: $Z=x+++y++;$

54

算術與邏輯運算式 混合計算

- $x=6, y=5$
- $(x \geq 5) \&\& y < 10$
- $!(x > 10) + (!y)$
- $!x > 10 + !y$

55

課堂演練

- 撰寫程式,比較下列運算式

```
double rate1=0.008;  
a=(rate1<0.01)&&(rate1>0.005)  
a=0.005<rate1<0.01
```

56