第十章

C語言的進階資料型態
（typedef,struct）

第十章 C語言的進階資料型態（typedef,struct）

- C語言允許程式設計師自訂資料型態。
- 自訂資料型態是只程式設計師可以組合多種基本資料型態，建構出更適用於程式邏輯的資料型態，以便設計程式。
- 我們將在這一章中，詳細說明這些自訂資料型態的語法與範例。

大綱

- 10.1 typedef型態定義
- 10.2 struct結構體
- 10.3 本章回顧

10.1 typedef型態定義

- C語言的基本資料型態其實分的很粗略，例如在數值方面，只分為整數與浮點數兩種基本資料型態。
- 而在實際的應用中，我們常常會希望將資料型態以更具意義的名字來命名，例如：要宣告成績變數的資料型態，就會想用score來宣告，要宣告價格的資料型態，就會想用price來宣告，而這兩種資料其實也是數值資料而已。

10.1 typedef型態定義

- C語言為了使得設計程式更加彈性，也更具可讀性，因此也允許我們這樣做，但必須先使用typedef來定義新的資料型態所對應的實際資料型態。換句話說，我們可以將某種資料型態的名稱改成另一個容易表達資料的識別字（別名），如此一來，在整個程式中，我們就能夠使用該別名來宣告資料型態了，typedef的語法格式如下：

10.1 typedef型態定義

- typedef語法：

```
typedef        資料型態        識別字（別名）；
```

- 語法說明：
  - (1)資料型態是識別字所對應的真實資料型態，它可以是C語言的基本資料型態，或其他已經定義過的自訂資料型態，或已經使用typedef定義過的別名。
  - (2)識別字（別名），一旦經由typedef定義之後，在程式中就可以使用該別名來宣告變數，而實際上，該別名將會被編譯器代替為原來的資料型態。

10.1 typedef型態定義

- 【範例】

```
typedef char * STRING;      /*  定義資料型態的別名  */
STRING str1="Book";
```

- 範例說明：
  - 經過定義後，STRING資料型態可視為指標字串資料型態char *的另一個別名，因此可以透過該型態宣告字串變數str1。

10.1 typedef型態定義

```
1
2   /*********************************
3       檔名:ch10_01.c
4       功能:typedef型態定義的練習
5    *********************************/
6   #include <stdio.h>
7   #include <stdlib.h>
8   typedef float score;
9   void main(void)
10  {
11   score stu[3],total,avg;
12   int i;
13    total=0;
14   for(i=1;i<=3;i++)
15   {
16     printf("請輸入第%d位同學的成績:",i);
17     scanf("%f",&stu[i-1]);
18     total=total+stu[i-1];
19   }
20   avg=total/3;
21   printf("平均成績=%.3f\n",avg);
22  }
```

10.1 typedef型態定義

- 執行結果：

```
請輸入第1位同學的成績:70
請輸入第2位同學的成績:88
請輸入第3位同學的成績:65
平均成績=74.333
```

  - 範例說明：
    - (1) 第8行，宣告了float資料型態的別名為score之後，程式中就可以使用這個識別字來宣告float資料型態的變數及陣列，例如第11行的stu[3]、total、avg的資料型態其實都是float。
    - (2) 明顯地，相對於使用float宣告變數或陣列，使用score來宣告變數或陣列，更能表達該變數或陣列內的資料是何種用途。

10.2 struct結構體

- 當我們的資料擁有很多項目時，在前面的範例中，我們會使用多維陣列來加以存放，但由於每一種資料可能擁有不同的資料型態，因此，我們有的時候無法單純使用多維陣列來加以存放資料，例如：我們要記錄全班同學的成績，每一位同學的資料為學號（字串資料型態）、計概成績（整數資料型態）、數學成績（整數資料型態）、英文成績（整數資料型態）、平均成績（浮點數資料型態）。

學號	數學	英文	計概	平均
"S9103501"	89	84	75	82.67
"S9103502"	77	69	87	77.67
"S9103503"	65	68	77	70.00

圖10-1 一筆資料可能需要使用多種資料型態

10.2 struct結構體

- C語言除了無法宣告不限資料型態的陣列之外，我們還很容易遇到一個問題，也就是假設我們強迫使用字串來儲存上述資料，仍舊無法使用之前學習的排序程式針對『計概』分數來做排序，因為陣列若經由遞增排序後會變成圖10-2左邊表格的狀況。

強迫使用相同資料型態(字串)儲存資料

排序

學號	計概	數學	英文	平均
"S9103501"	"65"	"84"	"75"	"82.67"
"S9103502"	"77"	"69"	"87"	"77.67"
"S9103503"	"89"	"68"	"77"	"70.00"

排序

學號	計概	數學	英文	平均
"S9103503"	"65"	"68"	"77"	"70.00"
"S9103502"	"77"	"69"	"87"	"77.67"
"S9103501"	"89"	"84"	"75"	"82.67"

圖10-2 強迫使用字串儲存資料仍舊有其他問題

10.2 struct結構體

- 在圖10-2左邊表格中，很明顯地如果僅僅將資料Array[i,1]排序的話，由於其他的資料不會跟著移動，因此『65』分並不是“S9103501”同學的計概成績，真正希望的排序後之結果應該是如圖10-2右邊表格的狀況。
- 當然我們可以在排序對調資料時，一併對調所有屬於同列的資料。但是這將導致程式更加複雜。此時，如果橫向的列可以一併移動該有多好。
- 為了解決上述的兩個問題，C語言允許程式設計師將多種資料變數，組合成一個獨特的資料型態，構成一個結構體，例如：我們可以將『學號』、『計概』、『數學』、『英文』、『平均』等成績合併為一個結構體，則可以解決上述問題，使得設計程式時更加方便，也使得程式更容易維護。

10.2.1 定義struct結構體及宣告結構體變數

- 在C語言中定義結構體，必須使用struct關鍵字，語法如下：
- struct（結構體）宣告語法：

```
struct 結構體資料型態名稱
{
        結構主體
};
```

10.2.1 定義struct結構體及宣告結構體變數

- 【範例1】

```
struct student
{
        char    stu_id[12];/*  學號  */
        int     ScoreComputer;/*  計概  */
        int     ScoreMath;/*  數學  */
        int     ScoreEng;/*  英文  */
        float   ScoreAvg;/*  平均成績  */
};
```

10.2.1 定義struct結構體及宣告結構體變數

- 範例說明：
  1. 定義結構體後，我們可以將結構體student視為一種新的資料型態，其中包含了stu_id、ScoreComputer、ScoreMath、ScoreEng、ScoreAvg等5個資料變數，如圖10-3所示。

  2. student結構體（新的資料型態）所宣告的變數，將會佔用28個位元組空間，如圖10-4示意。

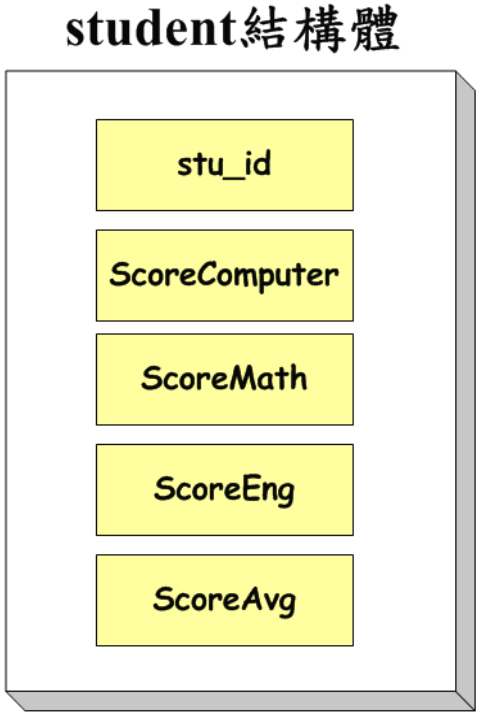

圖10-3 結構體示意圖

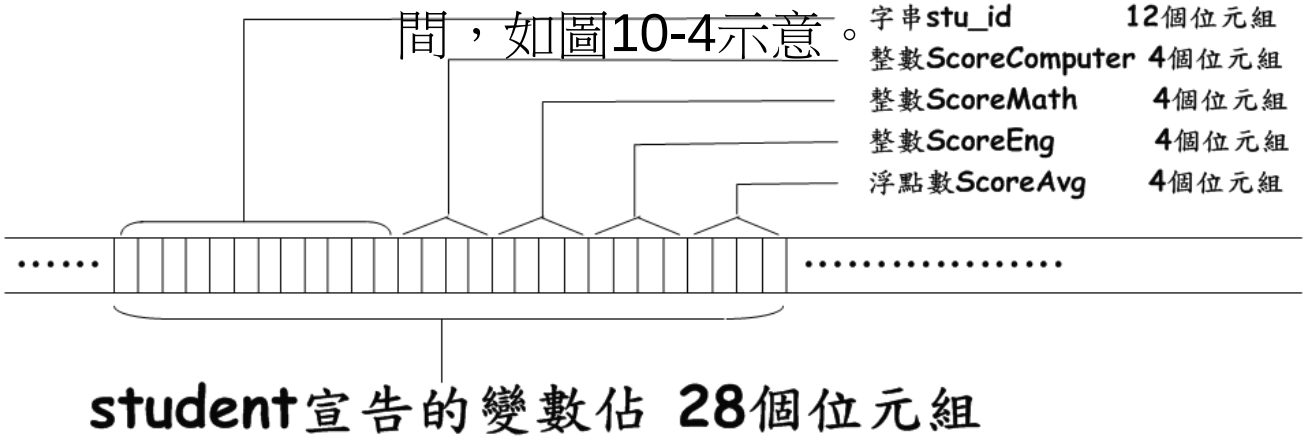

圖10-4 結構體在記憶體的示意圖

10.2.1 定義struct結構體及宣告結構體變數

- 當定義結構體之後，我們就可以將結構體當作是新的資料型態來宣告變數，而宣告語法則與基本資料型態類似：
- 使用結構體宣告變數語法：

```
struct   結構體名稱      變數或陣列名稱；
```

  - 語法說明：
    - 可以將結構體當做資料型態來宣告變數或陣列，並且在C++中，struct關鍵字可以省略（C語言不可省略）

10.2.1 定義struct結構體及宣告結構體變數

- 【範例1】

```
struct student John;
struct student IM[50];
```

  - 範例說明：
    - (1)John為一個student資料型態的變數。（在C++中，您也可以省略struct，宣告為student John;）
    - (2)IM[50]為大小50的陣列，每一個陣列元素的資料型態都是student型態。
    - (3)經由上述宣告後，編譯器將在記憶體中保留適當空間存放變數與陣列，如下圖示意：

10.2.1 定義struct結構體及宣告結構體變數

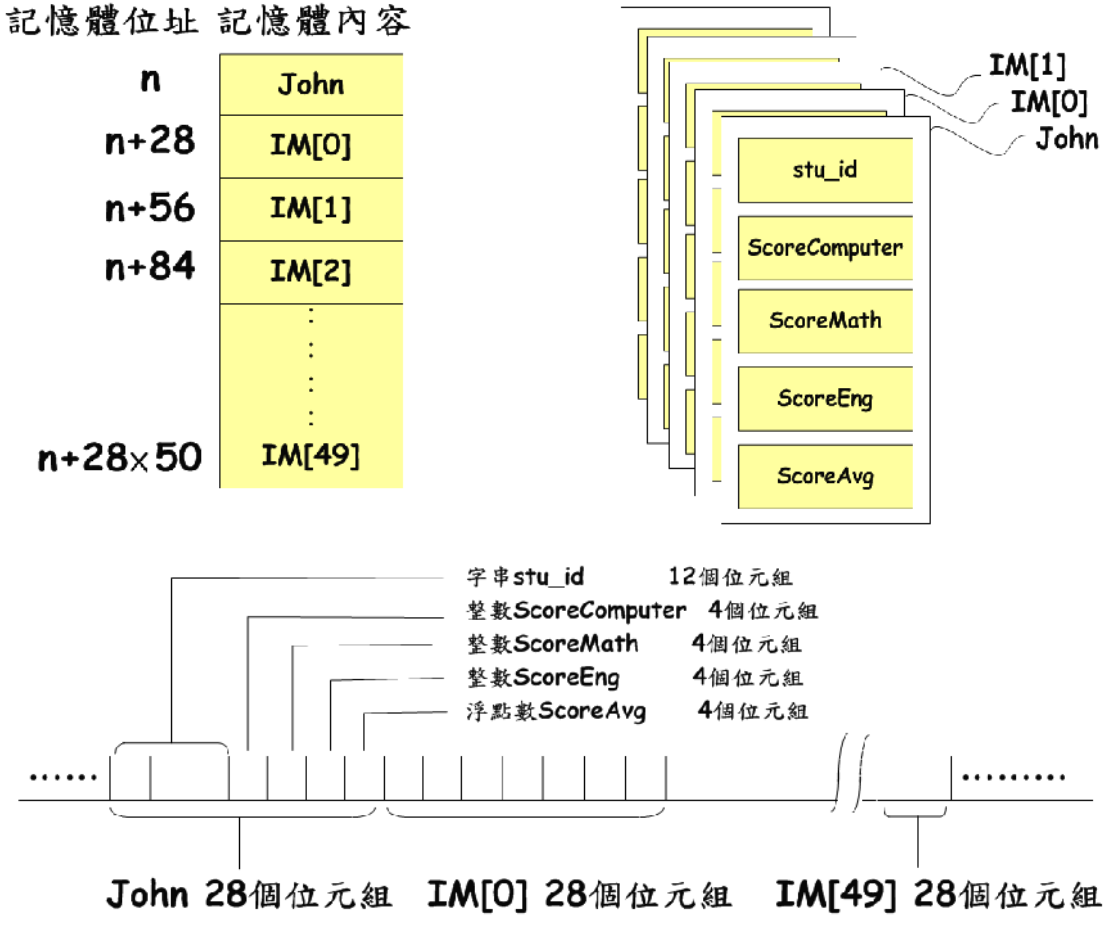

圖 10-5 結構體變數與陣列在記憶體中的狀況

10.2.2 存取結構體變數項目

- 當我們使用結構體宣告變數之後，若要存取結構體變數的某個資料項，可以用「.」與「->」符號來達成。兩種符號分別適用於不同時機如下：
  - (1)「.」：結構體為普通結構體變數。
  - 語法：

    結構體變數.資料項

  - 【範例】：設定學生John的數學成績及平均成績。

```
struct student John;
John.ScoreMath=86;
John.ScoreAvg=76.67;
```

10.2.2 存取結構體變數項目

- (2)「->」：結構體為指向結構體的指標變數。
- 語法：

  結構體指標變數->資料項

- 【範例】：透過指標設定學生John的數學成績及平均成績。

```
struct student John;
struct student *pJohn;
pJohn = &John;
pJohn->ScoreMath=86;
pJohn->ScoreAvg=76.67;
```

10.2.3 定義及宣告結構體的變形

- 定義及宣告結構體的變形（一）
  - 定義結構體變數，也可以和宣告結構體變數一併完成，如下語法：
  - 語法：

```
struct 結構體資料型態名稱
{
    結構主體
}結構體變數名稱;
```

  - 語法說明：
    - 上述語法不但可以定義一個結構體，也同時宣告了該結構體的一個變數實體。

10.2.3 定義及宣告結構體的變形

- 【範例】
- 範例說明：
  - X是結構體名稱，而Y則是結構體X宣告的變數。上述語法相當於下列語法：

```
struct X
{
   …
}Y;
```

```
struct X
{
   …
};
struct X Y;
```

10.2.4　結構體的結構體

- 結構體內包含許多項目，而每一個項目都可以是基本資料型態或之前宣告過的結構體（因為我們可以將已宣告過的結構體當作是一種自訂的新資料型態）。當結構體內的項目又是一個結構體時，也就是所謂的『結構體的結構體』。

- 相同於指標的指標存取方式，我們若要存取『結構體的結構體』的資料，只需要多加上數個『.』或『->』即可完成。

10.2.4 結構體的結構體

- 範例：

```
1  /*********************************
2      檔名:ch10_02.c
3      功能:struct定義及宣告結構體
4   *********************************/
5  #include <stdio.h>
6  #include <stdlib.h>
7  #include <string.h>
8  struct student
9  {
10   char   stu_id[12];
11   int    ScoreComputer;
12   int    ScoreMath;
13   int    ScoreEng;
14   float  ScoreAvg;
   };
```

10.2.4 結構體的結構體

```
15 void main(void)
16 {
17  int score[3][3]={{89,84,75},
18                   {77,69,87},
19                   {65,68,77}};
20  struct student IM[3];
21  struct student tempStu;
22  int i,Total;
23  strcpy(IM[0].stu_id,"S9103501");
24  strcpy(IM[1].stu_id,"S9103502");
25  strcpy(IM[2].stu_id,"S9103503");
26  for(i=0;i<3;i++)
27  {
28      Total=0;
29      IM[i].ScoreComputer=score[i][0];
30      IM[i].ScoreMath    =score[i][1];
31      IM[i].ScoreEng     =score[i][2];
32      Total=score[i][0]+score[i][1]+score[i][2];
33      IM[i].ScoreAvg=((float)Total)/3;
  }
```

10.2.4 結構體的結構體

```
35  printf("學號\t\t計概\t數學\t英文\t平均\n");
36  printf("-----------------------------------------------\n");
37  for(i=0;i<3;i++)
38  {
39       tempStu=IM[i];
40       printf("%s\t%d\t%d\t%d\t%.4f\n",\
41
42  tempStu.stu_id,tempStu.ScoreComputer,tempStu.ScoreMath,\
43                 tempStu.ScoreEng,tempStu.ScoreAvg);
44  }
45  /* system("pause"); */
}
```

10.2.4 結構體的結構體

- 執行結果：

```
學號          計概    數學    英文    平均
-----------------------------------------------
S9103501      89      84      75      82.6667
S9103502      77      69      87      77.6667
S9103503      65      68      77      70.0000
```

- 範例說明：
  - (1) 第8~14行，定義了一個結構體student，當中包含5個項目。
  - (2) 第20行，將結構體student視為新的資料型態，宣告一個1維陣列。
  - (3) 第21行，將結構體student視為新的資料型態，宣告一個變數tempStu。
  - (4) 第23~33行，設定IM[3]結構體陣列的每個元素項目資料。
  - (5) 第43~49行，顯示結構體陣列資料，我們透過tempStu來暫存每一個陣列元素。

10.2.5 結構體與函式

- 傳入結構體引數－傳值呼叫
- 範例：

```
1  /*********************************
2      檔名:ch10_03.c
3      功能:結構體引數
4   *********************************/
5  #include <stdio.h>
6  #include <stdlib.h>
7  #include <string.h>
8  struct student
9  {
10   char   stu_id[12];
11   int    ScoreComputer;
12   int    ScoreMath;
13   int    ScoreEng;
14   float  ScoreAvg;
   };
```

10.2.5 結構體與函式

```
15
16  void display(struct student);
17  void display(struct student tempStu)
18  {
19     printf("%s\t%d\t%d\t%d\t%.4f\n",\
20            tempStu.stu_id,tempStu.ScoreComputer,tempStu.ScoreMath,\
21            tempStu.ScoreEng,tempStu.ScoreAvg);
22  }
23  void main(void)
24  {
25   int score[3][3]={{89,84,75},
26                    {77,69,87},
27                    {65,68,77}};
28   struct student IM[3];
29   int i,Total;
30   strcpy(IM[0].stu_id,"S9103501");
     strcpy(IM[1].stu_id,"S9103502");
     strcpy(IM[2].stu_id,"S9103503");
```

10.2.5 結構體與函式

```
31  for(i=0;i<3;i++)
32   {
33       Total=0;
34       IM[i].ScoreComputer=score[i][0];
35       IM[i].ScoreMath    =score[i][1];
36       IM[i].ScoreEng     =score[i][2];
37       Total=score[i][0]+score[i][1]+score[i][2];
38       IM[i].ScoreAvg=((float)Total)/3;
39   }
40   printf("學號\t\t計概\t數學\t英文\t平均\n");
41   printf("-----------------------------------------------\n");
42   for(i=0;i<3;i++)
43   {
44     display(IM[i]);
45   }
46  }
```

10.2.5 結構體與函式

- 執行結果：

```
學號          計概    數學    英文    平均
-----------------------------------------------
S9103501      89      84      75      82.6667
S9103502      77      69      87      77.6667
S9103503      65      68      77      70.0000
```

- 範例說明：
  - (1) 第15行，宣告display函式，接受一個引數，該引數的資料型態是結構體student。
  - (2) 第16~22行，display函式的定義，顯示傳入結構體引數的各項資料。
  - (3) 第40行，呼叫display函式，並傳入一個結構體變數。

10.2.5 結構體與函式

- 傳入結構體指標－傳指標呼叫
- 範例：

```
1  /**********************
2      檔名:ch10_04.c
3      功能:傳遞結構體指標
4   ***********************/
5  #include <stdio.h>
6  #include <stdlib.h>
7  #include <string.h>
8  struct student
9  {
10   char   stu_id[12];
11   int    ScoreComputer;
12   int    ScoreMath;
13   int    ScoreEng;
14   float  ScoreAvg;
15 };
16 void display(struct student);
17 void BubbleSort(struct student
   *arr,int arr_index);
18  void display(struct student
   tempStu)
19 {
20
21 printf("%s\t%d\t%d\t%d\t%.4f\n",\
   tempStu.stu_id,tempStu.ScoreCompu
22 ter,tempStu.ScoreMath,\tempStu.Sc
   oreEng,tempStu.ScoreAvg);
23 }
24
```

10.2.5 結構體與函式

```
    void BubbleSort(struct student *arr,int arr_index)
25  {
26   int k,times,i;
27   struct student temp;
28
28   k=arr_index-1;
29   while(k!=0)
30   {
31    times=0;
32    for(i=0;i<=k-1;i++)
33    {
34     if(arr[i].ScoreComputer>arr[i+1].ScoreComputer)
35     {
36       temp=arr[i]; arr[i]=arr[i+1]; arr[i+1]=temp;
37       times=i;
38     }
39    }
40    k=times;
   }
}
```

10.2.5 結構體與函式

```
    void main(void)
40  {
41   int score[3][3]={{89,84,75},
42                    {77,69,87},
43                    {65,68,77}};
44   struct student IM[3],tempStu;
45   int i,Total;
46  strcpy(IM[0].stu_id,"S9103501");
47   strcpy(IM[1].stu_id,"S9103502");
48   strcpy(IM[2].stu_id,"S9103503");
49   for(i=0;i<3;i++)
50   {
51       Total=0;
52       IM[i].ScoreComputer=score[i][0];
53       IM[i].ScoreMath    =score[i][1];
54       IM[i].ScoreEng     =score[i][2];
55       Total=score[i][0]+score[i][1]+score[i][2];
56       IM[i].ScoreAvg=((float)Total)/3;
57   }
```

10.2.5 結構體與函式

```
58   printf("學號\t\t計概\t數學\t英文\t平均\t(依計概排序前)\n");
59   printf("-----------------------------------------------\n");
60   for(i=0;i<3;i++)
61     display(IM[i]);
62
63   BubbleSort(IM,3);
64   printf("學號\t\t計概\t數學\t英文\t平均\t(依計概排序後)\n");
65   printf("-----------------------------------------------\n");
66   for(i=0;i<3;i++)
67      display(IM[i]);
68 }
```

10.2.5 結構體與函式

- 執行結果：

```
學號        計概    數學    英文    平均      (依計概排序前)
-----------------------------------------------
S9103501    89      84      75      82.6667
S9103502    77      69      87      77.6667
S9103503    65      68      77      70.0000
學號        計概    數學    英文    平均      (依計概排序後)
-----------------------------------------------
S9103503    65      68      77      70.0000
S9103502    77      69      87      77.6667
S9103501    89      84      75      82.6667
```

- 範例說明：
  - (1) 第16行，宣告BubbleSort函式，接受結構體指標或結構體陣列名稱（可視為指標常數）。
  - (2) 第25~40行，Bubbluesort函式的定義，和前面章節介紹的差不多，只不過把排序依據改成每一個結構體。
  - (3) 由執行結果中，可以發現每次對調順序時，會將整個結構體元素對調，而不會只對調某個項目。

10.2.5 結構體與函式

- 回傳結構體
  - 函式既然可以接受結構體引數，同樣地，函式也可以回傳結構體變數，您只要在宣告及定義函式時，將回傳值型態指定為結構體即可（記得要加上struct）。
  - 語法：

```
struct 回傳的結構體名稱  函式名稱(引數串列)
{
   ….函式定義…
   return 結構體變數;
}
```

  - 語法說明：
    - 回傳的結構體名稱必須事先宣告過，同時您必須使用return來回傳該結構體變數。

10.2.5 結構體與函式

- 【範例】

```
struct student
{
  char   stu_id[12];
  int    ScoreComputer;
  int    ScoreMath;
  int    ScoreEng;
  float  ScoreAvg;
};

struct student cal(int i,int j)
{
 struct student X;
 ..................
return X;
}
```

- 範例說明：
  - cal()函式將回傳一個student結構體，在本範例中，X是回傳的結構體變數。

10.2.5 結構體與函式

- 結構體的視野
  - 結構體的視野（Scope）和變數的視野沒有太大的差別。
  - 前面的範例中，我們將結構體定義放在最前面，因此所有的函式都可以看見它。
  - 如果將結構體定義放到某個函式之中，則只有該函式可以看見該結構體；如果將結構體仍舊定義在函式之外，則只有在結構體定義之後的函式宣告可以看見該結構體。

10.2.6 struct結構體與typedef型態別名

- 【範例】

```
struct student
{
  char   stu_id[12];
  int    ScoreComputer;
  int    ScoreMath;
  int    ScoreEng;
  float  ScoreAvg;
};

typedef struct student Stu_Score;
```

  - 範例說明：
    - 經由定義別名後，您也可以使用Stu_Score來宣告結構體變數，例如：『Stu_Score John;』。

10.5 本章回顧

- 在本章中，我們介紹了C語言的進階資料結構，包含typedef、struct，其意義分述如下：
  - (1)資料型態別名：
    - 我們可以透過typedef為資料型態（包含基本資料型態、自訂資料型態）另外訂一個別名，以便表示資料的實質意義。
  - (2)結構體：
    - C語言允許程式設計師將多種資料變數，組合成一個獨特的資料型態，構成一個結構體，結構體分為兩種，分別由struct與union加以定義。struct定義的結構體稱為標準結構體；而union定義的結構體，稱為聯合結構體。union聯合結構體與struct結構體非

10.5　本章回顧

- 上述的各種進階資料結構，使得C程式設計更加具有彈性，也更有效率。在一般的資料結構（Data Structure）程式設計課程（例如：使用指標串列建立樹狀結構圖）以及中大型程式設計應用中，都常常必須搭配上述的各種進階資料結構來設計程式。

本章习题