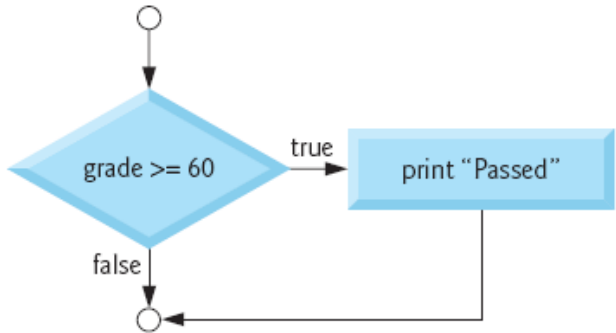


CHAPTER 2

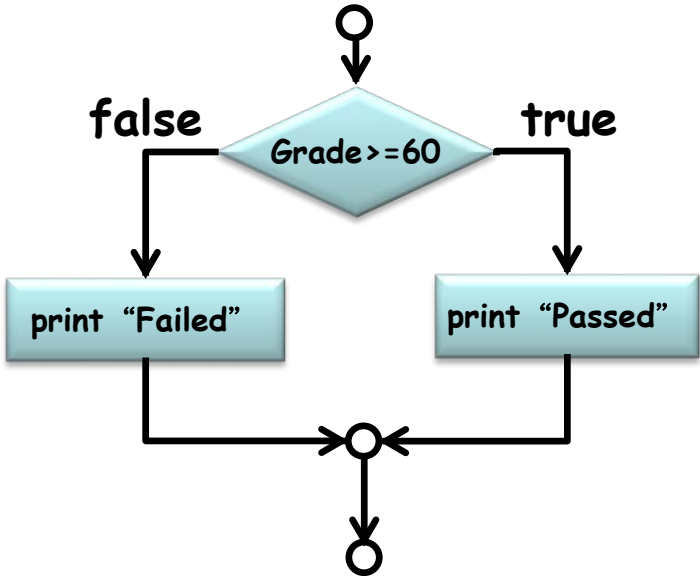


Structured Program (ch3)
Program Control (ch4)

- 設計結構化程式三步驟
 - Step1 演算法, Step2 流程圖, Step3 程式語言
- 問題: 如果考試高於60分, 就成績單列印過關

| Step1 演算法(Algorithm) 虛擬程式(Pseudocode) | Step2 流程圖 Flowchart | Step3 程式語言 Program |
|---|--|--|
| <p>如果成績大於60 列印過關</p> |  <pre> graph TD Start(()) --> Decision{grade >= 60} Decision -- true --> Print[print "Passed"] Decision -- false --> Exit(()) Print --> Exit </pre> <p>圖 3.2 單一選擇 if 敘述式的流程圖</p> | <pre> #include <stdio.h> #include <stdlib.h> int main(void) { int grade; printf("請輸入成績: "); scanf("%d", &grade); if (grade >= 60) { printf("Passed\n"); } system("pause"); return 0; } </pre> <div data-bbox="1586 1222 1856 1293" style="background-color: black; color: white; padding: 5px; text-align: center;"> 請輸入成績: 80 Passed </div> |

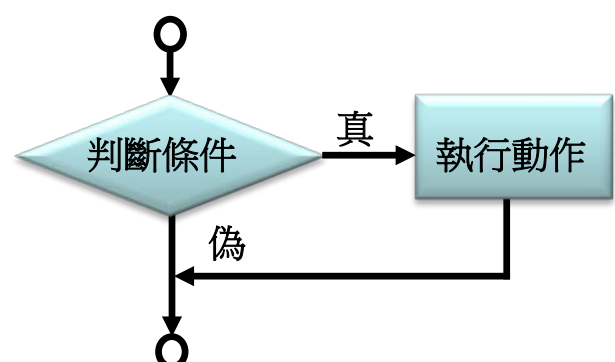
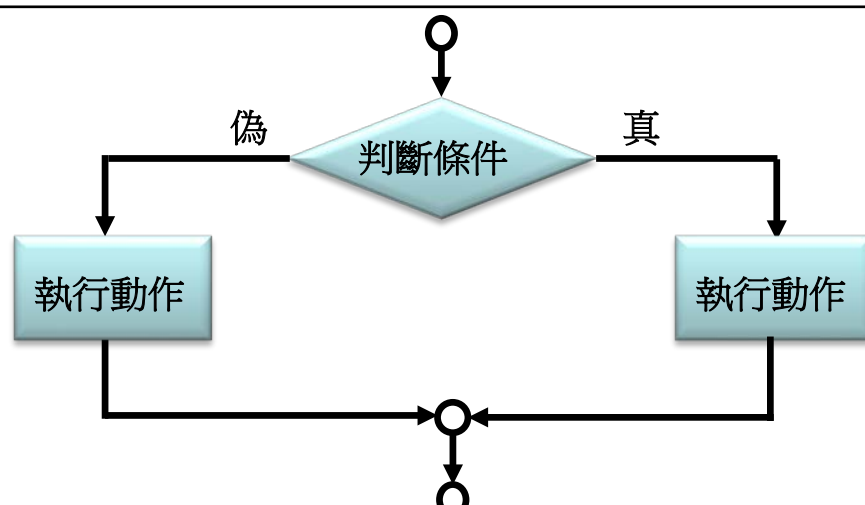
- 問題: 如果考試高於60分, 成績單列印過關, 否則列印失敗

| Step1 演算法(Algorithm) 虛擬程式(Pseudocode) | Step2 流程圖 Flowchart | Step3 程式語言 Program |
|---|---|--|
| <p>如果成績大於60 列印過關 否則 列印失敗</p> |  <pre> graph TD Start(()) --> Decision{Grade >= 60} Decision -- false --> Failed[print "Failed"] Decision -- true --> Passed[print "Passed"] Failed --> Join(()) Passed --> Join Join --> End(()) </pre> | <pre> #include <stdio.h> #include <stdlib.h> int main(void) { int grade; printf("請輸入成績: "); scanf("%d", &grade); if (grade >= 60) { printf("Passed\n"); } else { printf("Failed\n"); } system("pause"); return 0; } </pre> <div data-bbox="1632 1011 1903 1082" style="background-color: black; color: white; padding: 5px;"> 請輸入成績: 80 Passed </div> <div data-bbox="1632 1220 1903 1292" style="background-color: black; color: white; padding: 5px;"> 請輸入成績: 40 Failed </div> |

- C提供了與if...else敘述式十分類似的條件運算子 (?:)

| if...else | ?: |
|---|--|
| <pre>#include <stdio.h> #include <stdlib.h> int main(void) { int grade; printf("請輸入成績："); scanf("%d", &grade); if (grade >= 60) { printf("Passed\n"); } else { printf("Failed\n"); } system("pause"); return 0; }</pre> | <pre>#include <stdio.h> #include <stdlib.h> int main(void) { int grade; printf("請輸入成績："); scanf("%d", &grade); grade >= 60 ? printf("Passed\n") : printf("Failed\n"); system("pause"); return 0; }</pre> |

- If選擇條件敘述有兩種
 - 菱形代表判斷符號(decision symbol), 矩形代表動作符號(action symbol)

| C程式語法 | 流程圖 |
|---|---|
| If (判斷條件) {執行動作}; |  |
| If (判斷條件) {執行動作A}; else {執行動作B}; |  |

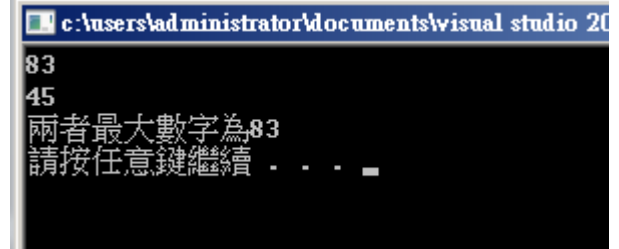
- 條件運算子 (?:)

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int n1;
    int n2;
    int max;

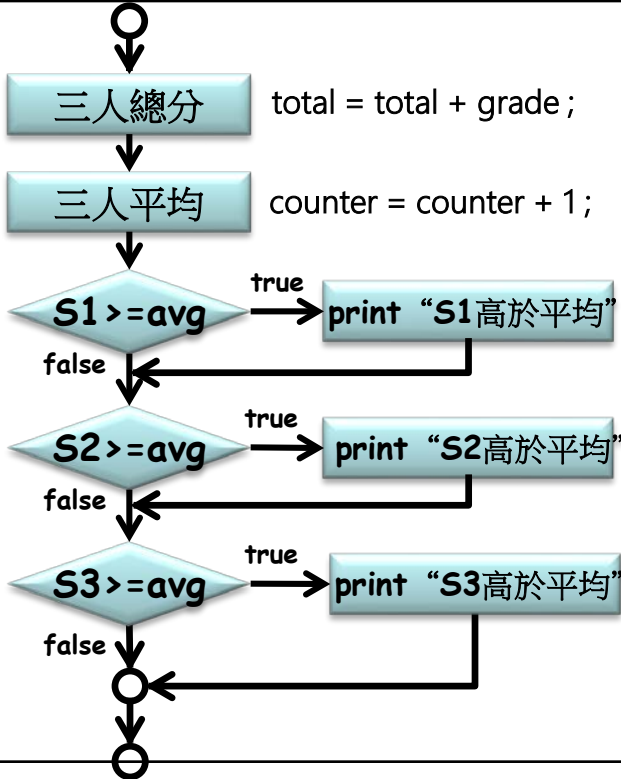
    scanf("%d%d", &n1, &n2);
    max = (n1 > n2) ? n1 : n2;
    printf("兩者最大數字為%d\n", max);

    system("pause");
}
```



```
c:\users\administrator\documents\visual studio 20...
83
45
兩者最大數字為83
請按任意鍵繼續 . . .
```

- 問題: 輸入班上三人成績, 顯示高於平均值的人
 - 使用if判斷敘述

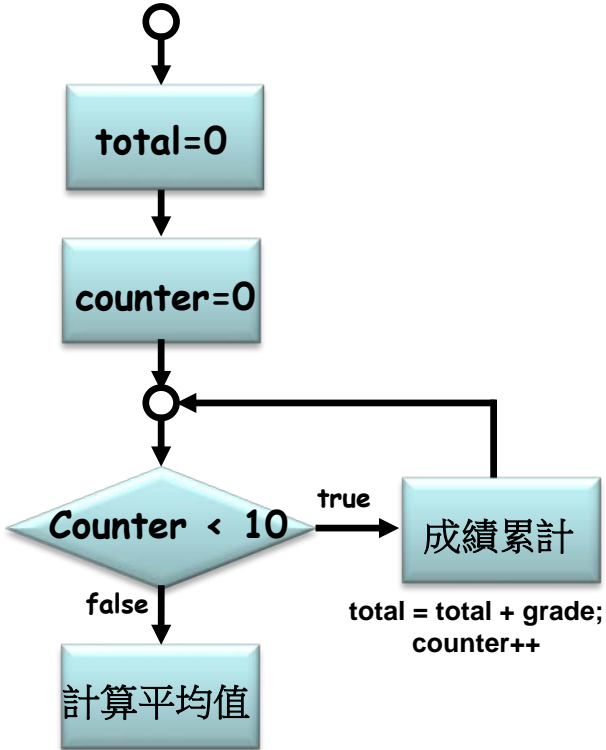
| Step1 演算法(Algorithm) 虛擬程式(Pseudocode) | Step2 流程圖 Flowchart | Step3 程式語言 Program |
|--|--|---|
| <ol style="list-style-type: none"> 1. 計算三人(S1, S2, S3)總分 2. 計算三人(S1, S2, S3)平均 3. 檢查每個學生(S1, S2, S3),如果高於平均值,列印高於平均 |  <pre> graph TD Start(()) --> Total[三人總分] Total --> Avg[三人平均] Avg --> S1{S1 >= avg} S1 -- true --> PrintS1[print "S1高於平均"] S1 -- false --> S2{S2 >= avg} S2 -- true --> PrintS2[print "S2高於平均"] S2 -- false --> S3{S3 >= avg} S3 -- true --> PrintS3[print "S3高於平均"] S3 -- false --> End(()) </pre> | <pre> 1 #include <stdio.h> 2 #include <stdlib.h> 3 int main(void) 4 { 5 int counter; 6 int grade; 7 int total; 8 int average; 9 int S1,S2,S3; 10 11 total=0; 12 counter=0; 13 14 printf("Enter grade: "); 15 scanf("%d", &grade); 16 S1=grade; 17 total=total+grade; 18 counter=counter+1; 19 20 printf("Enter grade: "); 21 scanf("%d", &grade); 22 S2=grade; 23 total=total+grade; 24 counter=counter+1; 25 26 printf("Enter grade: "); 27 scanf("%d", &grade); 28 S3=grade; 29 total=total+grade; 30 counter=counter+1; 31 32 average=total/counter; 33 34 if (S1>=average) 35 printf("S1高於平均\n"); 36 if (S2>=average) 37 printf("S2高於平均\n"); 38 if (S3>=average) 39 printf("S3高於平均\n"); 40 41 system("pause"); 42 return 0; 43 } </pre> <div data-bbox="1619 1106 1889 1268" style="background-color: black; color: white; padding: 5px;"> Enter grade : 80 Enter grade : 60 Enter grade : 40 S1高於平均 S2高於平均 </div> |

- C程式碼(使用if敘述):輸入班上三人成績,顯示高於平均值的人

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(void)
4 {
5     int counter;
6     int grade;
7     int total;
8     int average;
9     int S1,S2,S3;
10
11     total=0;
12     counter=0;
13
14     printf("Enter grade: ");
15     scanf("%d", &grade);
16     S1=grade;
17     total=total+grade;
18     counter=counter+1;
19
20     printf("Enter grade: ");
21     scanf("%d", &grade);
22     S2=grade;
23     total=total+grade;
24     counter=counter+1;
25
26     printf("Enter grade: ");
27     scanf("%d", &grade);
28     S3=grade;
29     total=total+grade;
30     counter=counter+1;
31
32     average=total/counter;
33
34     if (S1>=average)
35     {
36         printf("S1高於平均\n");
37     }
38     if (S2>=average)
39     {
40         printf("S2高於平均\n");
41     }
42     if (S3>=average)
43     {
44         printf("S3高於平均\n");
45     }
46
47     system("pause");
48     return 0;
49 }
```


```
Enter grade : 80
Enter grade : 60
Enter grade : 40
S1 高於平均
S2 高於平均
```


- 問題: 輸入班上十人成績, 算出平均值
 - 使用 **while** 迴圈敘述, 程式碼變簡潔

| Step1 演算法(Algorithm) 虛擬程式(Pseudocode) | Step2 流程圖 Flowchart | Step3 程式語言 Program |
|---|---|--|
| <ol style="list-style-type: none"> 初始化全班總分 <ul style="list-style-type: none"> total=0 初始化全班人數 counter=0 輸入班上成績十次, 計算總分 while(counter < 10) {成績累計, counter++} 計算平均值 |  <pre> graph TD Start([]) --> Total0[total=0] Total0 --> Counter0[counter=0] Counter0 --> LoopStart(()) LoopStart --> Decision{Counter < 10} Decision -- true --> Accumulate[成績累計 total = total + grade; counter++] Accumulate --> LoopStart Decision -- false --> Average[計算平均值] </pre> | <div style="display: flex; justify-content: space-between;"> <pre> 1 #include <stdio.h> 2 #include <stdlib.h> 3 4 int main(void) 5 { 6 int counter; 7 int grade; 8 int total; 9 int average; 10 11 total = 0; 12 counter = 0; 13 14 while (counter < 10) 15 { 16 printf("Enter grade:"); 17 scanf("%d", &grade); 18 total = total + grade; 19 counter = counter + 1; 20 } 21 22 average = total / counter; 23 printf("Class average is %d\n", average); 24 25 system("pause"); 26 27 return 0; 28 } </pre> <div style="background-color: black; color: white; padding: 5px; font-family: monospace;"> c:\users\administrator\document Enter grade:98 Enter grade:76 Enter grade:71 Enter grade:87 Enter grade:83 Enter grade:90 Enter grade:57 Enter grade:79 Enter grade:82 Enter grade:94 Class average is 81 請按任意鍵繼續 . . . </div> </div> |

- C程式碼(使用while敘述): 輸入班上十人成績, 算出平均值

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void)
5 {
6     int counter;
7     int grade;
8     int total;
9     int average;
10
11     total = 0;
12     counter = 0;
13
14     while (counter < 10)
15     {
16         printf("Enter grade:");
17         scanf("%d", &grade);
18         total = total + grade;
19         counter = counter + 1;
20     }
21
22     average = total / counter;
23     printf("Class average is %d\n", average);
24
25     system("pause");
26
27     return 0;
28 }
```



```
c:\users\administrator\documents\wi
Enter grade:98
Enter grade:76
Enter grade:71
Enter grade:87
Enter grade:83
Enter grade:90
Enter grade:57
Enter grade:79
Enter grade:82
Enter grade:94
Class average is 81
請按任意鍵繼續 - - -
```

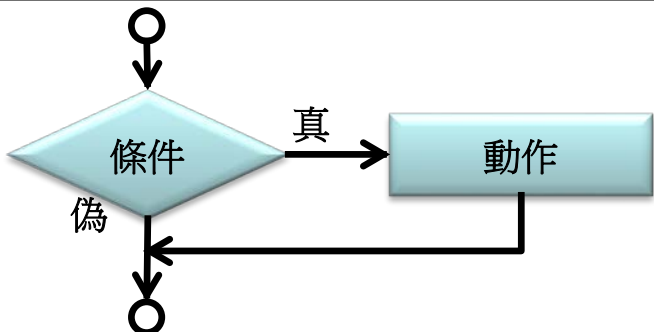
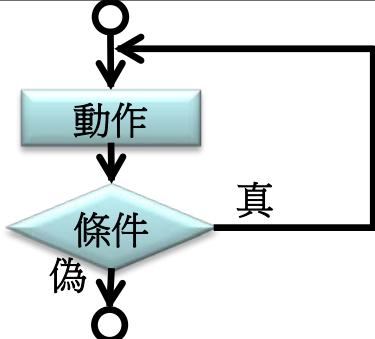
- C程式碼(使用 **while** 敘述): 輸入班上成績(人數可彈性,輸入者決定), 算出平均值
 - `while(grade != -1)` //彈性調整班上人數
 - `average = (float) total / counter;` //total強制轉換float資料型別, 除以counter(整數型別)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void)
5 {
6     int counter;
7     int grade;
8     int total;
9     float average;
10
11     total = 0;
12     counter = 0;
13
14     printf("Enter grade, -1 to end:");
15     scanf("%d",&grade);
16
17     while(grade != -1)
18     {
19         total = total + grade;
20         counter = counter + 1;
21
22         printf("Enter grade, -1 to end:");
23         scanf("%d", &grade);
24     }
```

```
25
26     if(counter != 0)
27     {
28         average = (float)total / counter;
29         printf("Class average is %.2f\n", average);
30     }
31     else
32     {
33         printf("No grades were entered\n");
34     }
35
36     system("pause");
37 }
```

```
c:\users\administrator\documents\visual studio 2010\Projects\ch2-10\Debug
Enter grade, -1 to end:90
Enter grade, -1 to end:85
Enter grade, -1 to end:74
Enter grade, -1 to end:63
Enter grade, -1 to end:88
Enter grade, -1 to end:97
Enter grade, -1 to end:-1
Class average is 82.83
請按任意鍵繼續 . . .
```

- **while**與**do...while**迴圈敘述
 - **do...while**重複敘述式十分類似**while**敘述式
 - **while**敘述式，迴圈本體(*statement*)執行之前就會檢查執行條件(*condition*)，條件成立才執行
 - **do...while**敘述式，先執行迴圈本體(*statement*)後，再檢查執行條件(*condition*)
 - 因此**do...while**迴圈本體至少會執行一次

| while | do...while |
|--|--|
| <pre>while (condition) { 迴圈本體(statement); }</pre> | <pre>do { 迴圈本體(statement); }while (condition);</pre> |
|  <pre> graph TD Entry(()) --> Cond{條件} Cond -- 真 --> Act[動作] Act --> Entry Cond -- 偽 --> Exit(()) </pre> |  <pre> graph TD Entry(()) --> Act[動作] Act --> Cond{條件} Cond -- 真 --> Entry Cond -- 偽 --> Exit(()) </pre> |

- HW: 韓信點兵, 共有10x10人
- (1)使用最外一圈畫出矩形
- (2)使用最外三圈畫出矩形
- (3)劃出上三角
- (4)劃出下三角
- (5)劃出X

```
c:\users\administrator\documents
<1>
XXXXXXXXXX
X-----X
X-----X
X-----X
X-----X
X-----X
X-----X
X-----X
X-----X
XXXXXXXXXX

請按任意鍵繼續 . . .
```

```
c:\users\administrator\documents
<2>
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXX----XXX
XXX----XXX
XXX----XXX
XXX----XXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX

請按任意鍵繼續 . . .
```

```
c:\users\administrator\documents
<3>
---XX---
--X-X--
-X--X--
X-----X
XXXXXXXXXX
-----
-----
-----
-----
-----

請按任意鍵繼續 . . .
```

```
c:\users\administrator\documents
<4>
XXXXXXXXXX
-X-----X
--X--X--
--X-X--
--XX---
-----
-----
-----
-----
-----

請按任意鍵繼續 . . .
```

```
c:\users\administrator\documents
<5>
X-----X
-X-----X
-X-----X
-X-X---
--XX---
--XX---
-X-X---
-X-----X
-X-----X
X-----X

請按任意鍵繼續 . . .
```

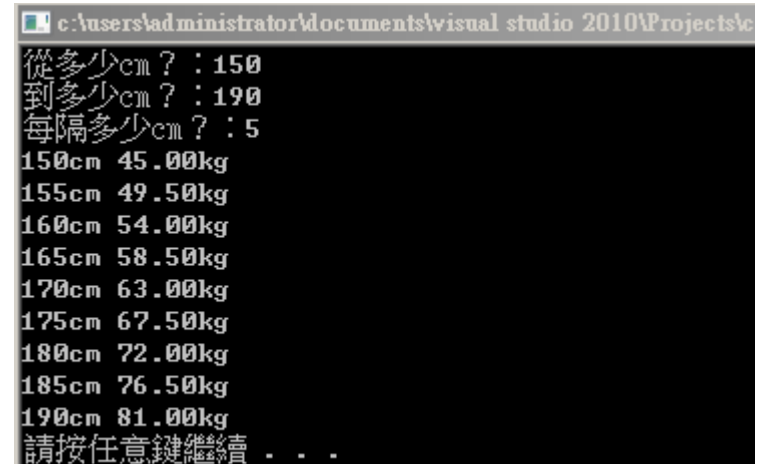
- 自動列印出**99**乘法表
- 使用者輸入兩個數字, 算出**99**乘法表結果

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|----|----|----|----|----|----|----|----|
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 2 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| 3 | 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 |
| 4 | 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 |
| 5 | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 |
| 6 | 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 |
| 7 | 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 |
| 8 | 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 |
| 9 | 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 |

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void)
5 {
6     int height;
7     int lower;
8     int upper;
9     int step;
10    double weight; |
11
12    printf("從多少cm? : ");    scanf("%d", &lower);
13    printf("到多少cm? : ");    scanf("%d", &upper);
14    printf("每隔多少cm? : ");    scanf("%d", &step);
15
16    for (height = lower; height <= upper; height += step)
17    {
18        weight = (height - 100) * 0.9;
19        printf("%dcm %.2fkg\n", height, weight);
20    }
21
22    system("pause");
23
24    return (0);
25 }
26

```

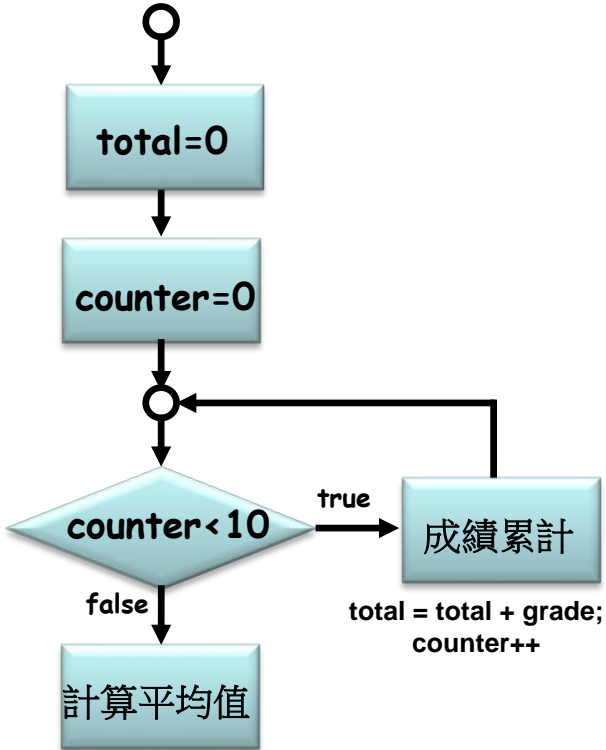


```

c:\users\administrator\documents\visual studio 2010\Projects\c
從多少cm? : 150
到多少cm? : 190
每隔多少cm? : 5
150cm 45.00kg
155cm 49.50kg
160cm 54.00kg
165cm 58.50kg
170cm 63.00kg
175cm 67.50kg
180cm 72.00kg
185cm 76.50kg
190cm 81.00kg
請按任意鍵繼續 . . .

```

- 問題: 輸入班上十人成績, 算出平均值
 - 使用for迴圈敘述

| Step1 演算法(Algorithm) 虛擬程式(Pseudocode) | Step2 流程圖 Flowchart | Step3 程式語言 Program |
|---|---|---|
| <ol style="list-style-type: none"> 初始化全班總分 <ul style="list-style-type: none"> <code>total=0</code> 全班人數計算 <code>counter</code> 輸入班上成績十次, 計算總分 <code>for(counter =1; counter <=10; counter++)</code> <code>{成績累計}</code> 計算平均值 |  <pre> graph TD Start([]) --> Total0[total=0] Total0 --> Counter0[counter=0] Counter0 --> LoopStart(()) LoopStart --> Decision{counter < 10} Decision -- true --> Accumulate[成績累計 total = total + grade; counter++] Accumulate --> LoopStart Decision -- false --> Avg[計算平均值] </pre> | <pre> 1 #include <stdio.h> 2 #include <stdlib.h> 3 int main(void) 4 { 5 int counter; 6 int grade; 7 int total; 8 int average; 9 10 total=0; 11 12 for (counter=0; counter<10; counter++) 13 { 14 printf("Enter grade: "); 15 scanf("%d", &grade); 16 total=total+grade; 17 } 18 19 average=total / counter; 20 21 printf("Class average is %d\n",average); 22 23 system("pause"); 24 return 0; 25 } </pre> <div data-bbox="1586 511 1901 835" style="background-color: black; color: white; padding: 5px;"> Enter grade : 98 Enter grade : 76 Enter grade : 71 Enter grade : 87 Enter grade : 83 Enter grade : 90 Enter grade : 57 Enter grade : 79 Enter grade : 82 Enter grade : 94 Class average is 81 </div> |

- 問題: 輸入幾月份, 程式顯示季節
 - 使用 **if** 判斷敘述

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 |
4 int main(void)
5 {
6     int month;
7
8     printf("請輸入月份:");
9     scanf("%d", &month);
10
11     if(month == 3 || month == 4 || month == 5)
12     {
13         puts("這個月份是春天。");
14     }
15     else if(month == 6 || month == 7 || month == 8)
16     {
17         puts("這個月份是夏天。");
18     }
19     else if(month == 9 || month == 10 || month == 11)
20     {
21         puts("這個月份是秋天。");
22     }
23     else if(month == 12 || month == 1 || month == 2)
24     {
25         puts("這個月份是冬天。");
26     }
```

```
27     else
28     {
29         puts("您輸入的月份不存在!\a");
30     }
31
32     system("pause");
33     return (0);
34 }
```



```
c:\users\administrator\documents\visual studio 2010\Proj
請輸入月份:11
這個月份是秋天。
請按任意鍵繼續 . . .
```

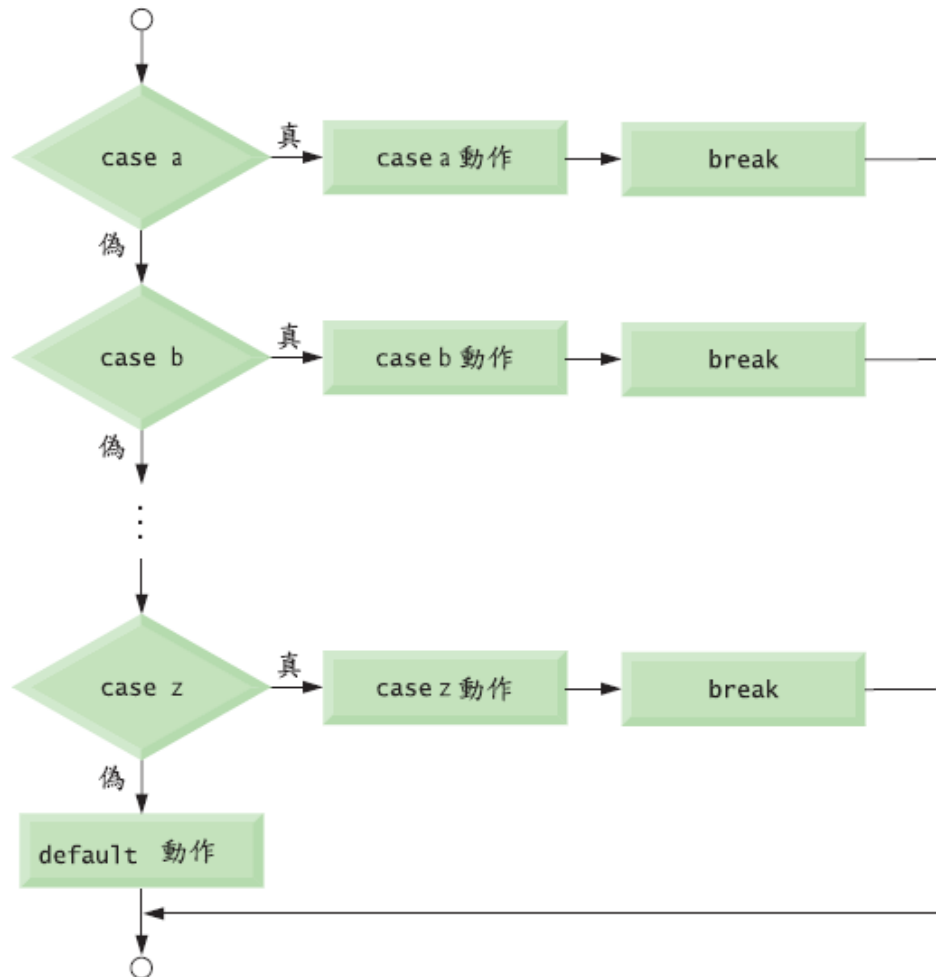
- 問題: 輸入幾月份, 程式顯示季節
 - 使用 **switch** 判斷敘述

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(void)
4 {
5     int month;
6
7     printf("請輸入月份: ");
8     scanf("%d", &month);
9
10    switch (month)
11    {
12        case 3:
13        case 4:
14        case 5:
15            puts("這個月份是春天");
16            break;
17        case 6:
18        case 7:
19        case 8:
20            puts("這個月份是夏天");
21            break;
22        case 9:
23        case 10:
24        case 11:
25            puts("這個月份是秋天");
26            break;
27        case 12:
28        case 1:
29        case 2:
30            puts("這個月份是冬天");
31            break;
32        default:
33            puts("你輸入的月份不存在! \a");
34            break;
35    }
36
37    system("pause");
38    return (0);
39 }
```

- **switch**敘述與**if**敘述

- **if**敘述和**switch**敘述都是讓程式流程做選擇的敘述，故合稱為**選擇敘述**（**selection statement**）。
- **switch**敘述較具整體性，故使用**單一運算式之值**讓程式流程做選擇的時候，選用**switch**敘述會比較好。

switch flowchart (switch流程圖)



switch程式語法

```

switch(判斷條件)
{
    case 'a':
        case a 動作;
        break; /跳出switch/

    case 'b':
        case b 動作;
        break; /跳出switch/

    .
    .
    case 'z':
        case z 動作;
        break; /跳出switch/

    default:
        default 動作;
        break; /跳出switch/
}
  
```

- break 和 continue

break

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void)
5 {
6     int x;
7
8     for(x=1; x<=10; x++)
9     {
10         if(x==5)
11         {
12             break; // break loop only if x is 5
13         }
14
15         printf("%d ", x);
16     }
17
18     printf("\nBroke out of loop at x == %d\n", x);
19     system("pause");
20     return 0;
21 }

```

```

c:\users\administrator\documents\visual studio
1 2 3 4
Broke out of loop at x == 5
請按任意鍵繼續 . . .

```

continue

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void)
5 {
6     int x;
7
8     for(x=1; x<=10; x++)
9     {
10         if(x==5)
11         {
12             continue; // skip remaining code in loop body
13         }
14
15         printf("%d ", x);
16     }
17
18     printf("\nUsed continue to skip printing the value 5\n");
19     system("pause");
20     return 0;
21 }

```

```

c:\users\administrator\documents\visual studio 2010\Projects
1 2 3 4 6 7 8 9 10
Used continue to skip printing the value 5
請按任意鍵繼續 . . .

```

- 遞增和遞減運算子

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(void)
4 {
5     int c;
6
7     c=5;
8     printf("%d\n",c);
9     printf("%d\n",c++); /* print 5 then postincrement */
10    printf("%d\n\n",c); /* print 6 */
11
12    c=5;
13    printf("%d\n",c);
14    printf("%d\n",++c); /* preincrement then 6 */
15    printf("%d\n",c); /* print 6 */
16
17    system("pause");
18    return 0;
19 }

```

```

5
5
6
5
6
6

```

| 運算子 | 範例運算式 | 說明 |
|-----|-------|------------------------|
| ++ | ++a | 先將 a 遞增 1，再以 a 的新值進行運算 |
| ++ | a++ | 以 a 目前的值進行運算，再將 a 遞增 1 |
| -- | --b | 先將 b 遞減 1 再以 b 的新值進行運算 |
| -- | b-- | 以 b 目前的值進行運算，再將 b 遞減 1 |

圖 3.12 遞增和遞減運算子

- 邏輯AND運算子
 - `a && b` 若a及b均非零，則其值為1
 - ex: `if (gender == 1 && age >= 65)`
`++seniorFemales;`

| 運算式 1 | 運算式 2 | 運算式 1 && 運算式 2 |
|-------|-------|----------------|
| 0 | 0 | 0 |
| 0 | 非零 | 0 |
| 非零 | 0 | 0 |
| 非零 | 非零 | 1 |

圖 4.13 && (邏輯 AND) 運算子的真值表

- 邏輯OR運算子
 - `a || b` 若a或b非零時，則其值為1
 - `if (semesterAverage >= 90 || finalExam >= 90)
printf("Student grade is A\n");`

| 運算式 1 | 運算式 2 | 運算式 1 運算式 2 |
|-------|-------|----------------|
| 0 | 0 | 0 |
| 0 | 非零 | 1 |
| 非零 | 0 | 1 |
| 非零 | 非零 | 1 |

圖 4.14 邏輯 OR (||) 運算子的真值表

- 常見的if(=)錯誤, if(==)正確
- 舉例來說，假設我們想要寫的是

```
if ( payCode == 4 ) //正確
    printf( "You get a bonus!" );
```

but we accidentally write

```
if ( payCode = 4 ) //錯誤, 但是compiler通常無法檢查
    printf( "You get a bonus!" );
```

- 第二個if敘述式if (payCode = 4)是錯的敘述式，任何非零的數值都會解釋為「真」，因此if敘述式的條件將永遠為真，不管那個人payCode是多少，都會拿到獎金！

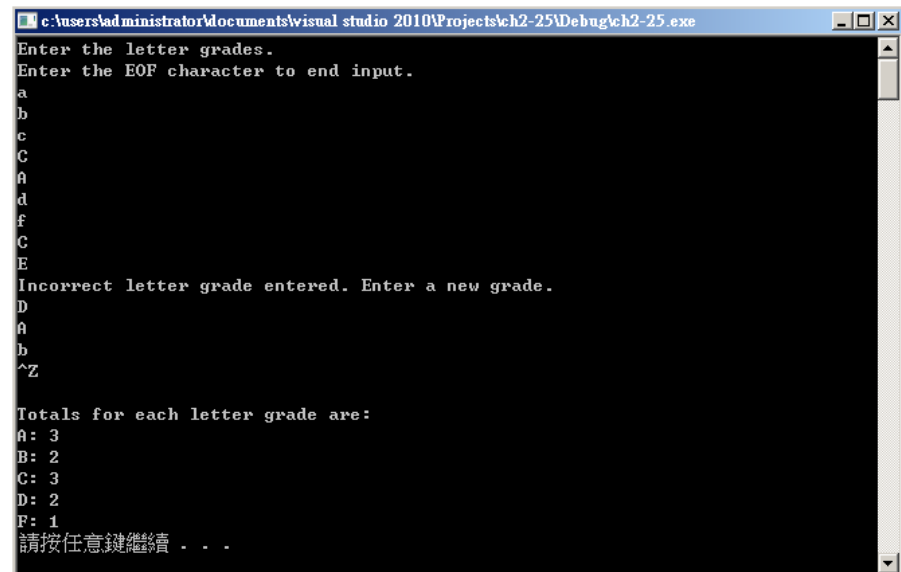
- Switch範例程式: 班上分成A, B, C, D 四個level,計算各別人數

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void)
5 {
6     int grade;
7     int aCount = 0;
8     int bCount = 0;
9     int cCount = 0;
10    int dCount = 0;
11    int fCount = 0;
12
13    printf("Enter the letter grades.\n");
14    printf("Enter the EOF character to end input.\n");
15
16    while( (grade = getchar() ) != EOF)
17    {
18        switch(grade)
19        {
20            case 'A':
21            case 'a':
22                ++aCount;
23                break;
```

```
24        case 'B':
25        case 'b':
26            ++bCount;
27            break;
28        case 'C':
29        case 'c':
30            ++cCount;
31            break;
32        case 'D':
33        case 'd':
34            ++dCount;
35            break;
36        case 'F':
37        case 'f':
38            ++fCount;
39            break;
40        case '\n':
41        case '\t':
42        case ' ':
43            break;
44        default:
45            printf("Incorrect letter grade entered.");
46            printf(" Enter a new grade.\n");
47            break;
```

- Switch範例程式: 計算各A, B, C, D 四個level各別人數

```
48     }
49 }
50
51 printf("\nTotals for each letter grade are:\n");
52 printf("A: %d\n", aCount);
53 printf("B: %d\n", bCount);
54 printf("C: %d\n", cCount);
55 printf("D: %d\n", dCount);
56 printf("F: %d\n", fCount);
57
58 system("pause");
59 return 0;
60 }
```

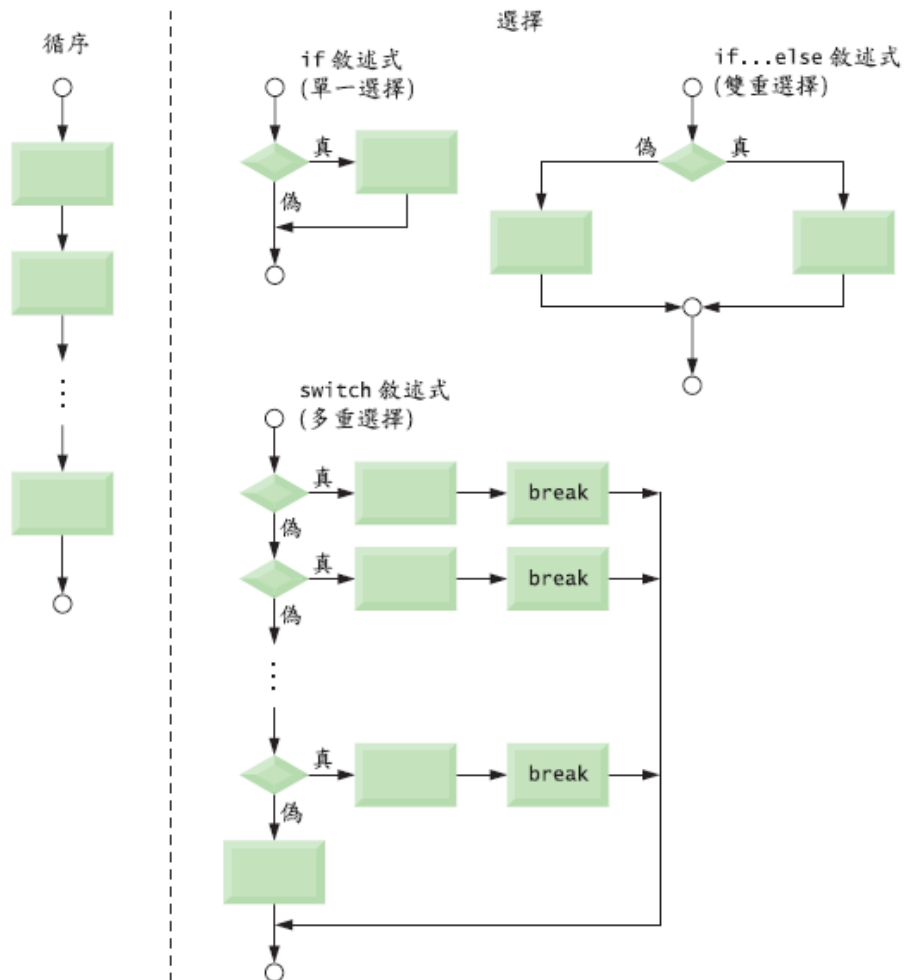


```
c:\users\administrator\documents\visual studio 2010\Projects\ch2-25\Debug\ch2-25.exe
Enter the letter grades.
Enter the EOF character to end input.
a
b
c
C
A
d
f
C
E
Incorrect letter grade entered. Enter a new grade.
D
A
b
^Z

Totals for each letter grade are:
A: 3
B: 2
C: 3
D: 2
F: 1
請按任意鍵繼續 . . .
```

- `getchar`函式
 - 來自`<stdio.h>`
 - 從鍵盤讀進一個字元，並將此字元存放到整數變數`grade`當中。
- `grade = getchar()`
 - 這個指定運算式的值，就是由`getchar()`傳回並且指定給變數`grade`的字元。
- `while((grade = getchar()) != EOF)`
 - `EOF`是定義在`<stdio.h>`前置檔裡的一個整數常數
 - 指定式`grade = getchar()`的值會拿來與`EOF`（代表`end of file`）比較
 - 以`EOF`當成警示值(`EOF`通常為`-1`)
 - 如果指定給`grade`的值等於`EOF`，則程式便會停止輸入資料

- 結構化程式正確流程圖



- 結構化程式正確流程圖

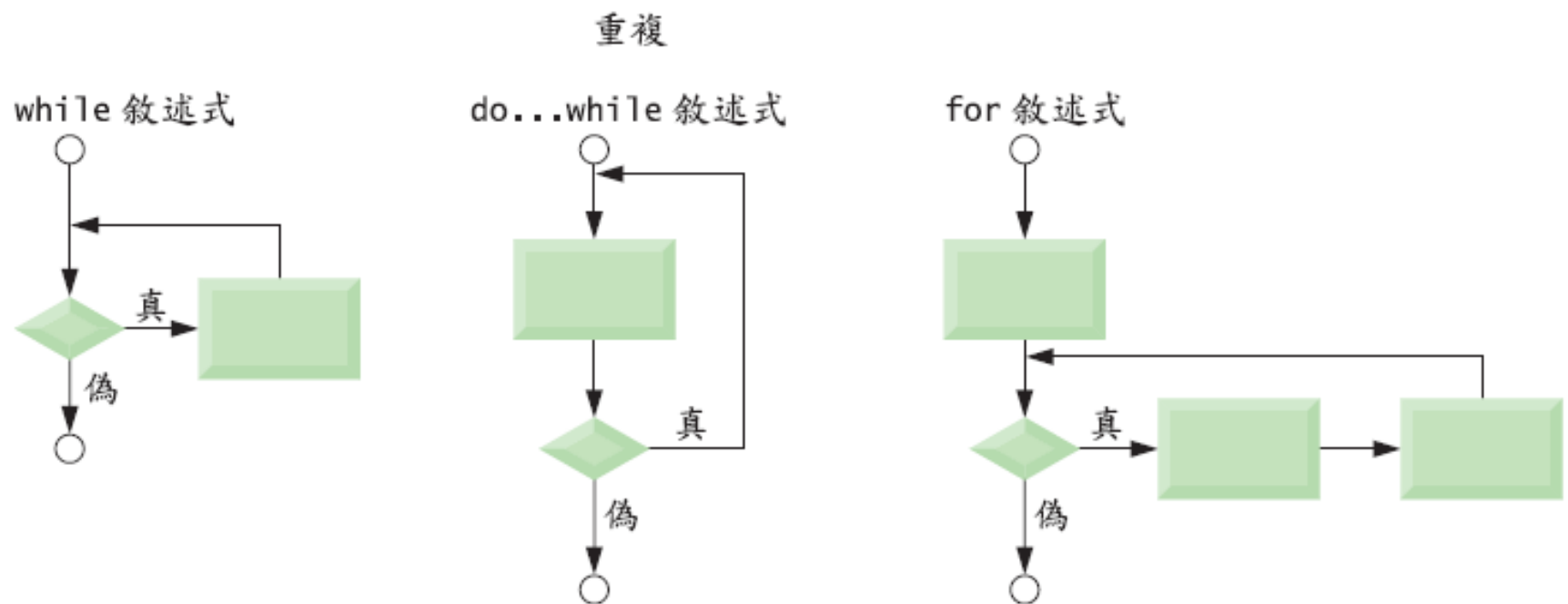


圖 4.17 C 的單一入口／單一出口之循序，選擇和重複敘述式

- 結構化程式正確流程圖

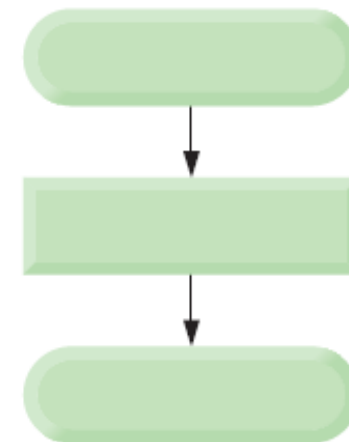


圖 4.19 最簡單的流程圖

結構化程式建構規則

- 1) 從 " 最簡單的流程圖 "(圖 4.19) 著手。
- 2) 任何矩形 (動作) 都可以用兩個連續的矩形 (動作) 來取代。
- 3) 任何矩形 (動作) 可以由任何的 control 敘述式 (循序 , if, if...else, switch, while, do...while 或 for) 來取代。
- 4) 規則 2 和規則 3 可隨你喜歡地運用。

圖 4.18 結構化程式建構規則
NTUT MMS LAB

- 結構化程式正確流程圖

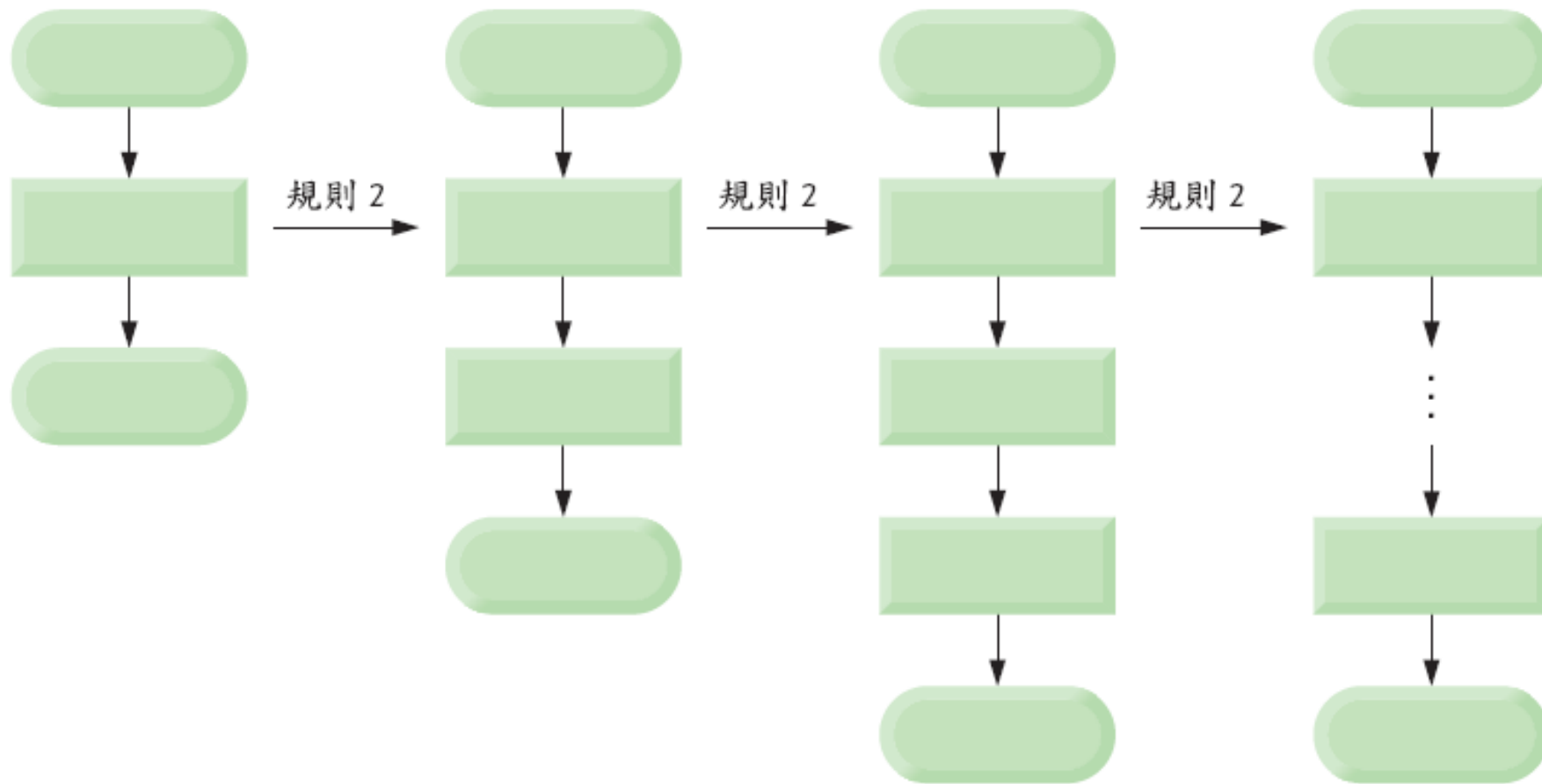


圖 4.20 對最簡單的流程圖重複運用圖 4.18 的規則 2

- 結構化程式正確流程圖

結構化程式建構規則

- 1) 從 " 最簡單的流程圖 "(圖 4.19) 著手。
- 2) 任何矩形 (動作) 都可以用兩個連續的矩形 (動作) 來取代。
- 3) 任何矩形 (動作) 可以由任何的敘述式 (循序 , if, if...else, switch, while, do...while 或 for) 來取代。
- 4) 規則 2 和規則 3 可隨你喜歡地運用。

圖 4.18 結構化程式建構規則

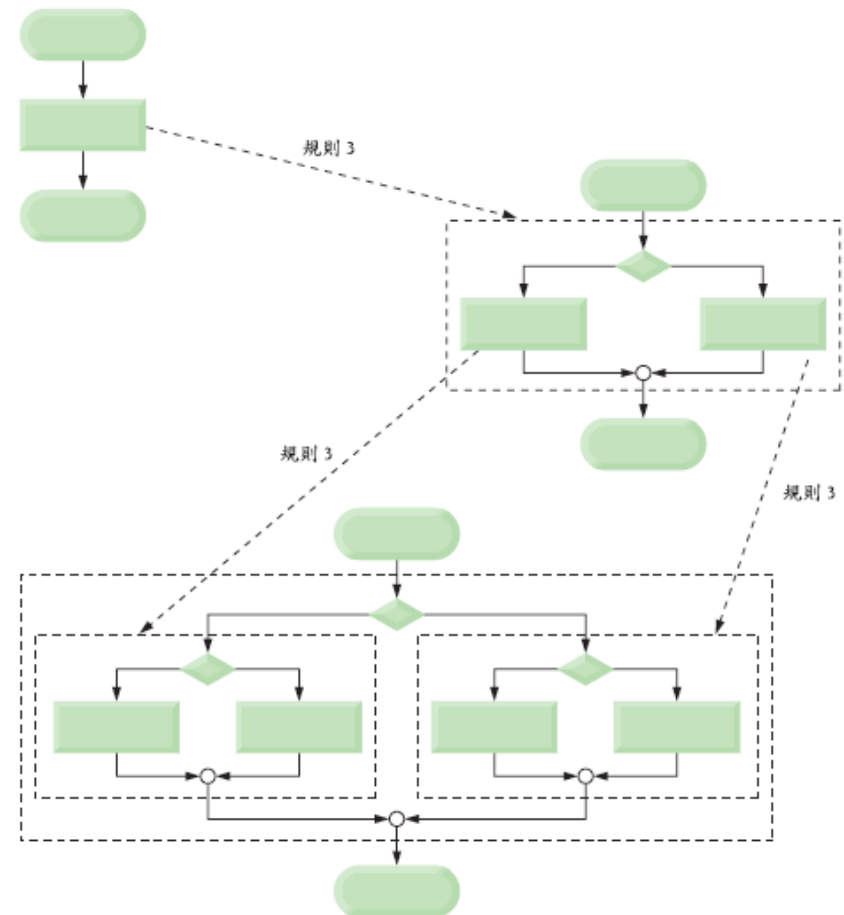
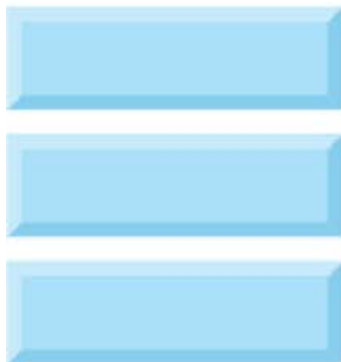


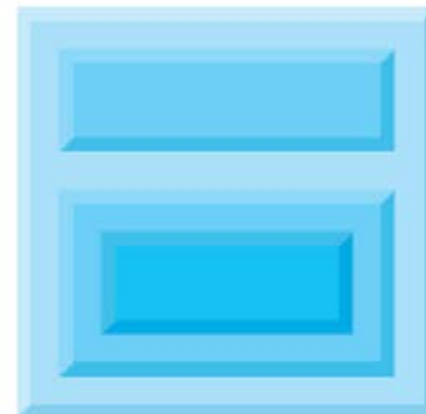
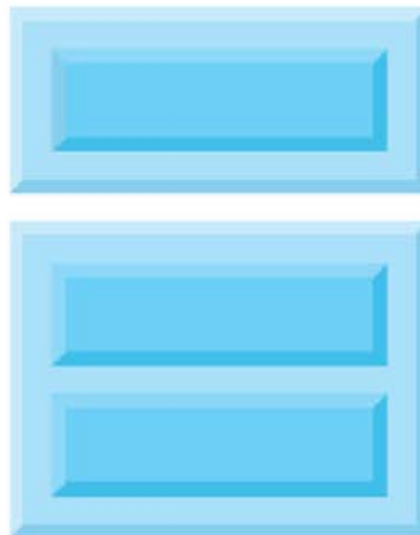
圖 4.21 對最簡單的流程圖重複運用圖 4.18 的規則 3

- 結構化程式正確流程圖

堆疊式的建構區塊



巢狀的建構區塊



- 非結構化錯誤流程圖

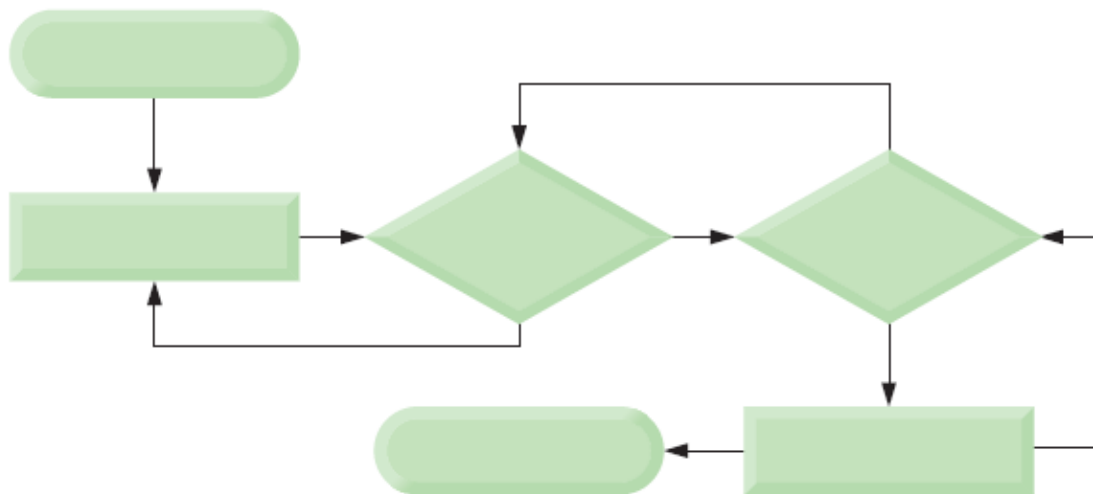


圖 4.23 非結構化流程圖

重疊的建構區塊
(違反結構化的程式設計)

