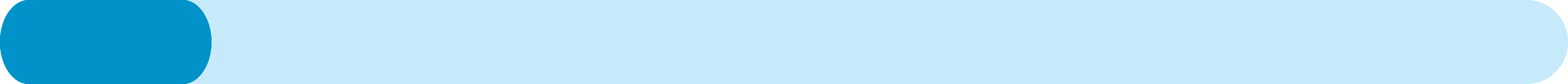
# 07



## 結構與類別

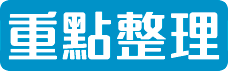
* 1. 結構
  2. 結構陣列
  3. 結構指標
  4. 類�

程式設計實習



7-1

結構



1. 結構(Structure)是一種**使用者自行建立的資料型態**，由一個以上的成員組成，成員可以是整數、浮點數或陣列等資料型態。
2. c 不允許函式做為結構成員，但 c++允許，因此 c++的結構和類別(class)很相似，都可以達到物件導向的功能。
3. 建立結構需使用 **struct** 關鍵字，語法如下：

struct 結構名稱｛成員 1；

成員 2；

:

}；

|  |  |  |
| --- | --- | --- |
| struct body{ | /\* | 結構名稱：body \*/ |
| int id; | /\* | 成員 1：整數 \*/ |
| char name[10]; | /\* | 成員 2：字元陣列 \*/ |
| char sex; | /\* | 成員 3：字元 \*/ |
| float height, weight; | /\* | 成員 4,5：浮點數 \*/ |
| }; | | |

1. 建立結構後，即可宣告該結構型態的變數，稱為結構變數，語法如下：

struct 結構名稱 結構變數名稱；

struct body s1; /\* 宣告結構變數 s1，型態為 body 結構 \*/

宣告結構變數時，可以同時指定成員的初始值，

struct 結構名稱 結構變數名稱 ｛成員 1 初始恒， 成員 2 初始恒， …}；

struct body s2 = {111002, "王小明", 'M', 171.5, 65.8}; /\* 宣告結構變數 s2，指定成員初始恒 \*/

宣告結構變數時，c 語言需加上 struct 關鍵字，c++則可以省略。

1. 建立結構和宣告結構變數可以在一行程式敘述句完成，如下列程式碼，同時建立 body 結構和宣告結構變數 s1、s2，

|  |
| --- |
| struct body{ |
| int id; |
| char name[10]; |
| char sex; |
| float height, weight; |
| }**s1, s2={111002, "王小明", 'M', 171.5, 65.8}**; |

#### 7-2

Chapter 7 結構與類�

1. 使用「.j運算子存取結構成員，

|  |
| --- |
| s1.id = 111001; |
| strcpy(s1.name, "王小美"); /\* 使用 strcpy()函式指定字串恒 \*/ |
| s1.sex = 'F'; |
| s1.height = 165.5; |
| s1.weight = 49.9; |
|  |
| cout << "學號：" << s2.id << endl; |
| cout << "姓名：" << s2.name << endl ; |
| cout << "性�：" << s2.sex << endl ; |
| cout << "身高：" << s2.height << endl ; |
| cout << "體重：" << s2.weight << endl ; |

***01 *** 

保健室阿姨為了記錄新 入學時的身體健康狀況，請電腦社社長莎莎幫忙設計程式，莎莎決定使用結構來記錄每個人的基本資料(性別、身高與體重)，而除了記錄基本資料外，同時也能輸出 BMI 值，保健室阿姨提供 2 筆測試資料，供莎莎驗證程式的執行結果。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 學號 | 姓名 | 性� | 身高 | 體重 |
| 11101 | 王小美 | F | 165.5 | 49.9 |
| 11102 | 王小明 | M | 171.5 | 65.8 |

程式碼：

|  |
| --- |
| 1 #include <iostream> |
| 2 #include <cmath> |
| 3 |
| 4 using namespace std; |
| 5 |
| 6 struct body{ |
| 7 int id; |
| 8 char name[10]; |
| 9 char sex; |
| 10 float height, weight; |
| 11 }; |
| 12 |
| 13 int main() { |
| 14 struct body s1; |
| 15 struct body s2; |

#### 7-3

程式設計實習

|  |
| --- |
| 16 |
| 17 cout << "輸入第 1 位同學的基本資料：" << endl; |
| 18 cin >> s1.id; |
| 19 cin >> s1.name; |
| 20 cin >> s1.sex; |
| 21 cin >> s1.height; |
| 22 cin >> s1.weight; |
| 23 cout << endl; |
| 24 |
| 25 cout << "輸入第 2 位同學的基本資料：" << endl; |
| 26 cin >> s2.id; |
| 27 cin >> s2.name; |
| 28 cin >> s2.sex; |
| 29 cin >> s2.height; |
| 30 cin >> s2.weight; |
| 31 cout << endl; |
| 32 |
| 33 cout << "學號：" << s1.id << endl; |
| 34 cout << "姓名：" << s1.name << endl; |
| 35 cout << "性別：" << s1.sex << endl; |
| 36 cout << "BMI：" << s1.weight/pow(s1.height/100, 2) << endl; |
| 37 cout << "---" << endl; |
| 38 |
| 39 cout << "學號：" << s2.id << endl; |
| 40 cout << "姓名：" << s2.name << endl; |
| 41 cout << "性別：" << s2.sex << endl; |
| 42 cout << "BMI：" << s2.weight/pow(s2.height/100, 2) << endl; |
| 43 } |

程式碼說明：

|  |  |
| --- | --- |
| 行數 | 說明 |
| 6 ~ 11 | 建立 body 結構，具備 5 個成員。 |
| 14， 15 | 宣告 body 結構變數，分別為 s1 和 s2。 |
| 17 ~ 31 | 使用 c++提供的 cin 物件，由使用者輸入資料並指定給結構成員。 |
| 33 ~ 42 | 使用 c++提供的 cout 物件，輸出結構變數 s1 和 s2 的內容。  BMI 的計算公式為 BMI  體重(kg)  身高(m)2 |

#### 7-4

Chapter 7 結構與類�



( B ) 1. cIc++語言中允許使用者自己定義資料型態，該資料型態可以包含整數、浮點數等基本型態，這種自己定義的新資料型態，稱為：

(A)陣列 (B)結構 (c)種類 (D)集合。

( D ) 2. 在 c 語言中要建立一個結構(structure)，需使用哪一個關鍵字？

(A)define (B)array (c)class (D)struct。

( B ) 3. 在 cIc++語言中，要存取結構的成員，要使用哪一個運算子？ (A)「-j (B)「.j (c)「.j (D)「~j。

( A ) 4. 關於 cIc++語言的結構(structure)型態，下列何者錯誤？ (A)結構的成員不可以是陣列

(B)是程式設計人員自行定義的資料型態

(c)要在程式中建立結構，需使用 struct 關鍵字

(D)要存取結構變數的成員，需使用「.j運算子。

( c ) 5. 東東使用 c 語言建立一結構 device，程式碼如下，請問宣告一個 device 結構變數，所需的記憶體 間為何？ (A)16Byte (B)4Byte (c)20Byte (D)32Byte。

struct device { int gps\_x; int gps\_y; int gps\_z; double value;

};

( D ) 6. 承上題，要宣告一個 device 結構的結構變數 a，下列何者是正確的語法？ (A)struct a = device， (B)a = struct(device)， (c)device struct a， (D)struct device a，。

( A ) 7. 承上題，要指定結構變數 a 的成員 value 為 3.14，下例何者是正確的語法？ (A)a.value = 3.14， (B)a->value = 3.14 (c)a(value) = 3.14， (D)a..value = 3.14。



1.(B)

2.(D)

3.(B)

4.(A)

5.(c)

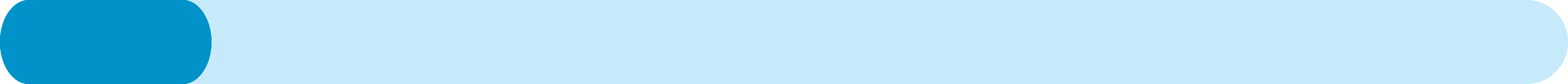
6.(D)

7.(A)

5. device 結構包含 3 個 int(4 Byte)成員，1 個 double(8 Byte)成員。

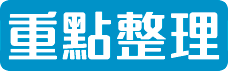
打 表示有詳解

程式設計實習



7-2

結構陣列



1. 陣列的元素為結構資料型態，稱為結構陣列(Structure Array)。
2. 宣告結構陣列的語法如下，

struct 結構名稱 陣列名稱［元素個數］；

struct body student[3]; /\* 宣告具有 3 個元素的結構陣列 student，每個元素都是—個範例 01 的 body

結構 \*/

1. 使用陣列索引的方式存取結構的成員，

|  |
| --- |
| student[0].height = 171.5; |
| student[1].height = 178.2; |
| student[2].height = 166.8; |
|  |
| for (int i=0; i<3; i++){ /\* 使用for 迴圏走訪結構陣列 \*/ |
| cout << "身高：" << student[i].height << endl; |
| } |

***02 *** 

在 P7-3 範例 01 中，莎莎幫保健室阿姨設計一支程式，記錄同學的基本資料，但是要為全校每一位同學都宣告一個結構變數，是非常不智的做法，因此莎莎決定使用結構陣列來改寫程式，同時能提高程式的可讀性且使程式更易於維護。

程式碼：

|  |  |
| --- | --- |
| 1 | #include <iostream> |
| 2 | #include <cmath> |
| 3 | |
| 4 | #define STUDENT\_NUM 1000 |
| 5 | |
| 6 | using namespace std; |
| 7 | |
| 8 | struct body{ |
| 9 | int id; |
| 10 | char name[10]; |
| 11 | char sex; |
| 12 | float height, weight; |
| 13 | }; |

Chapter 7 結構與類�

|  |  |
| --- | --- |
| 14 | |
| 15 | int main() { |
| 16 | struct body s[STUDENT\_NUM]; |
| 17 | |
| 18 | for (int i=0; i<STUDENT\_NUM; i++){ |
| 19 | cout << "輸入學號：" << endl; |
| 20 | cin >> s[i].id; |
| 21 | cin >> s[i].name; |
| 22 | cin >> s[i].sex; |
| 23 | cin >> s[i].height; |
| 24 | cin >> s[i].weight; |
| 25 | cout << endl; |
| 26 | } |
| 27 | |
| 28 | for (int i=0; i<STUDENT\_NUM; i++){ |
| 29 | cout << "學號：" << s[i].id << endl; |
| 30 | cout << "姓名：" << s[i].name << endl; |
| 31 | cout << "性�：" << s[i].sex << endl; |
| 32 | cout << "BMI：" << s[i].weight/pow(s[i].height/100, 2) << endl; |
| 33 | cout << "---" << endl; |
| 34 | } |
| 35 | } |

程式碼說明：

|  |  |
| --- | --- |
| 行數 | 說明 |
| 4 | 用 前置處 理 器 指令 **＃define** 建 立 識別 字 STUDENT一NUM ，在程式內只要 出 現  STUDENT一NUM，就用 1000 取代。 |
| 8 ~ 13 | 建立 body 結構，具備 5 個成員。 |
| 16 | 宣告一個有 1000 個元素的結構陣列 s，每個元素皆為 body 結構型態。 |
| 18 ~ 26 | 使用 for 迴圈走訪結構陣列 s，將資料存入。 |
| 28 ~ 34 | 使用 for 迴圈走訪結構陣列 s，將資料輸出。  BMI 的計算公式為 BMI  體重(kg)  身高(m)2 |

程式設計實習



( c ) 1. cIc++的程式設計人員可以自行建立結構資料型態，並宣告結構變數來儲存資料，下列關於結構的說明，何者是錯誤的？

(A)結構陣列是多個相同結構型態的變數的集合 (B)結構指標指向的資料型態是結構

(c)結構內的成員必須是相同的資料型態 (D)結構陣列使用索引的方式存取元素。

( B ) 2. 亮亮使用 c 語言的結構陣列來記錄好朋友的電話號碼，他寫了一段程式碼如下，請問要存取最後一位朋友的電話，哪一個語法才是正確的？



(A) friend[0].num， (B) friend[19].num， (c) phone[0].num， (D) phone[19].num，

struct phone{ int num;

char name[10];

};

int main() {

struct phone friend[20];

}

( B ) 3. 承上題，陣列 friend[ ]總共需要多大的記憶體 間？ (A)14 Byte (B)280 Byte (c)140 Byte (D)200 Byte。

( D ) 4. 關於下列程式碼片段的描述，何者正確？

(A)結構 player 擁有 3 個成員

(B) nba[ ]是一個結構陣列，每個元素都是一個 player 結構型態

(c)一個 player 結構的變數會佔用記憶體 64Bbyte 的大小

(D)以上皆是。

struct player{ char name[30]; char team[30]; int pay;

};

int main() {

struct player nba[100]; strcpy(nba[0].name,"LeBron James"); strcpy(nba[0].team,"Lakers"); nba[0].pay = 3922;

}

Chapter 7 結構與類�

( B ) 5. 下列程式片段的輸出結果為何？ (A)10 (B)9 (c)0 (D)20。

struct S{

int x; int y;

};

int main() {

struct S A[10];

for (int i=0;i<10;i++){ A[i].x = i;

A[i].y = A[i].x;

}

cout << A[0].x + A[9].y << endl;

}



1.(c)

2.(B)

3.(B)

4.(D)

5.(B)

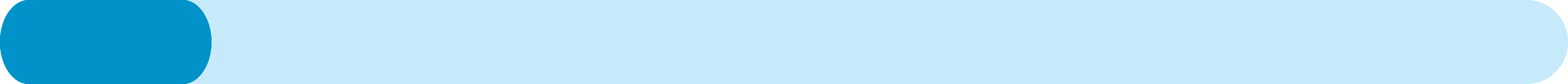
3. phone 結構所需的記憶體大小為 14 Byte，結構陣列 friend[20]有 20 個元素，每個元素都是 phone 結構，

故所需的記憶體大小為 20x14 Byte=280 Byte。

5. 結構陣列 A[]的每個元素都是 S 結構，使用迴圈走訪 A[]，將每個元素的成員 x 和 y 都指定為陣列索引

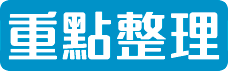
值，故 A[0].x + A[9].y = 0 + 9 = 9。

打 表示有詳解



7-3

結構指標



* + 1. 如同整數指標指向整數變數，**結構指標會指向結構變數**，宣告結構指標的語法如下，

struct 結構名稱 ＊指標名稱；

|  |
| --- |
| struct date{ /\* 建立 date 結構，存放年丶月丶曰資料 \*/ |
| int year; |
| int month; |
| int day; |
| }; |
|  |
| struct date birthday; /\* 宣告— date 結構型態的**結構變數**，名稱為 birthday \*/ |
| struct date \*ptr; /\* 宣告— date 結構型態的**結構指標**，名稱為 ptr \*/ |

* + 1. 使用取址運算子「＆j，將結構指標指向結構變數，

ptr = &birthday; /\* 結構指標 ptr 指向結構變數 birthday \*/

程式設計實習

程式執行後，ptr 與 birthday 在記憶體的關係如下，







1. 存取結構指標指向的結構變數成員，有 2 種方式：
   1. 使用取**值運算子「＊j**

|  |
| --- |
| (\*ptr).year = 2004; |
| (\*ptr).month = 8; |
| (\*ptr).day = 15; |
|  |
| cout << "Your birthday is " << (\*ptr).year << "-" << (\*ptr).month << "-" << (\*ptr).day << endl; |

##### **使用**指標運算子「-＞j

|  |
| --- |
| ptr->year = 2004; |
| ptr->month = 8; |
| ptr->day = 15; |
|  |
| cout << "Your birthday is " << ptr->year << "-" << ptr->month << "-" << ptr->day << endl; |

1. 在函式中傳遞結構型態的資料，可以採用傳值呼叫或傳址呼叫，
   1. 傳值呼叫，參數是結構變數，不會改變來源結構變數的內容。

|  |
| --- |
| 1. void setDate(struct date); /\* 函式原型宣告，規範參數需為date 結構型態 \*/ |
| 2. |
| 3. void setDate(struct date d){ /\* 函式定義(實作)，參數 d 為結構變數 \*/ |
| 4. d.year = 2022; |
| 5. cout << d.year << endl; /\* 輸出 2022 \*/ |
| 6. } |
| 7. |
| 8. int main(){ |
| 9. struct date birthday; |
| 10. birthday.year = 2000; |
| 11. setDate(birthday); /\* 呼叫 setDate()，引數為結構變數 birthday \*/ |
| 12. cout << birthday.year << endl; /\* 輸出 2000 \*/ |
| 13. } |

Chapter 7 結構與類�

程式執行後，變數在記憶體內的變化如下，第 9 行，宣告一 date 型態的結構變數 birthday



第 11 行，呼叫函式 setDate(birthday)，採傳值呼叫，等同於執行

struct date d = birthday;



第 4 行，修改結構變數 d 的成員，不會改變結構變數 birthday 的內容

第 6 行，函式結束，d 消失，返回第 12 行繼續執行



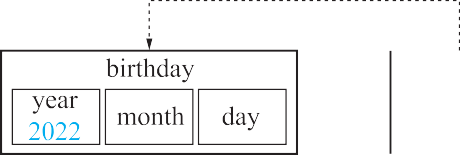
* 1. 傳址呼叫，參數是結構指標，指向來源結構變數，會直接改變來源結構變數的內容。

|  |
| --- |
| 1. void setDate(struct date\*); /\* 函式原型宣告，規範參數需為結構指標 \*/ |
| 2. |
| 3. void setDate(struct date \*ptr){ /\* 函式定義(實作)，參數ptr 為結構指標 \*/ |
| 4. ptr->year = 2022; |
| 5. cout << ptr->year << endl; /\* 輸出 2022 \*/ |
| 6. } |
| 7. |
| 8. int main(){ |
| 9. struct date birthday; |
| 10. birthday.year = 2000; |
| 11. setDate(&birthday); /\* 呼叫 setDate()，引數為結構變數 birthday 的位址 \*/ |
| 12. cout << birthday.year << endl; /\* 輸出 2022 \*/ |
| 13. } |

程式執行後，變數在記憶體內的變化如下，第 9 行，宣告一 date 型態的結構變數 birthday



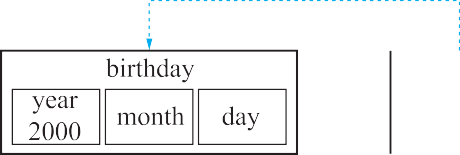
程式設計實習



第 11 行，呼叫函式 setDate(＆birthday)，採傳址呼叫，等同於執行

struct date \*ptr = &birthday;





第 4 行，改變 ptr 指向的結構變數內容，等於直接改變來源結構變數 birthday 的內容



第 6 行，函式結束，ptr 消失，返回第 12 行繼續執行



***03 *** 

巧虎爸爸正在學習 c++語言的結構，他想到可以建立一個日期結構，用來記錄巧虎的 日，同時使用 2 個函式來設定和顯示 日，巧虎媽媽也提醒爸爸，結構資料在函式中可以用傳值

或傳址的方式傳遞，可以嘗試用不同的方式設計(巧虎的 日是 5 月 5 日)。



程式碼：

|  |
| --- |
| 1 #include <iostream> |
| 2 using namespace std; |
| 3 |
| 4 struct date{ /\* 建立 date 結構 \*/ |
| 5 int month; |
| 6 int day; |
| 7 }; |
| 8 |
| 9 void printDate(struct date); |
| 10 void setDate(struct date\*, int, int); |
| 11 |
| 12 void printDate(struct date d){ |
| 13 cout << "Your birthday is " << d.month << "/" << d.day; |

Chapter 7 結構與類�

|  |
| --- |
| 14 } |
| 15 void setDate(struct date \*ptr, int mm, int dd){ |
| 16 ptr->month = mm; |
| 17 ptr->day = dd; |
| 18 } |
| 19 |
| 20 int main() { |
| 21 struct date birthday; |
| 22 struct date \*ptr; |
| 23 |
| 24 ptr = &birthday; |
| 25 |
| 26 setDate(ptr, 5, 5); |
| 27 printDate(birthday); |
| } |

程式碼說明：



|  |  |
| --- | --- |
| 行數 | 說明 |
| 4~7 | 建立 date 結構，有 2 個成員。 |
| 9 | 宣告 printDate( )函式原型，規範參數必須是 date 結構型態。 |
| 10 | 宣告 setDate( )函式原型，規範 3 個參數分別是指向 date 結構的結構指標和 2 個整數。 |
| 12~14 | 實作 printDate( )函式。 |
| 15~18 | 實作 setDate( )函式，參數 ptr 是指向 date 結構的結構指標，使用「->j運算子存取結構成員，會直接改變來源結構的成員。 |
| 21 | 宣告一 date 結構型態的結構變數 birthday。 |
| 22 | 宣告一指向 date 結構型態的結構指標 ptr。 |
| 24 | 使結構指標 ptr 指向結構變數 birthday。 |
| 26 | 呼叫 setDate( )函式，傳入的引數為結構指標 ptr(brithday 結構變數的位址)和 2 個整數。 |
| 27 | 呼叫 printDate( )函式，傳入的引數為 brithday 結構變數。 |



( B ) 1. 小健宣告一個結構指標 p 並指向一自訂結構 s，要使用該指標存取結構的成員，需使用哪一個運算子？ (A)「.j (B)「->j (c)「.j (D)「-j。

程式設計實習

( D ) 2. 下列有關 c 語言運算子的說明，哪一個選項不正確？

* + 1. 「＆＆j是邏輯運算子
    2. 「.j是結構成員運算子，用來存取結構的成員 (c)結構指標經由「->j運算子存取結構成員

(D)「＆j是取值運算子。

( B ) 3. 小新在 c++語言中宣告一個結構變數 data 和結構指標 pData，要使 pData 指向 data，需使用下列哪一個語法？



(A)pData = data， (B)pData = ＆data， (c)pData = ＊data (D)pData->data。

( B ) 4. 承上題，結構變數 data 有一個成員 id，下列哪一個語法，無法將 id 設定為 100？

(A) (＊pData).id = 100， (B) pData.id = 100，

(c) data.id = 100， (D) pData->id = 100，。 ( D ) 5. 一程式片段如下，下列何者的值與其它 3 個不同？

(A)(＊p).x (B)box.x (c)p->x (D)p.x。

struct device{ /\* 建立 device 結構 \*/ int x;

int y;

};

struct device box; struct device \*p; p = &box;

( c ) 6. 下列程式片段執行後的輸出為何？ (A)20 10 (B)10 10 (c)10 20 (D)20 20。

struct ST{

int a; int b;

};

void F(struct ST m){ int tmp;

tmp = m.a;

m.a = m.b;

m.b = tmp;

}

int main() {

struct ST m;

m.a = 10;

m.b = 20;

F(m);

printf("%d %d", m.a, m.b);

}

Chapter 7 結構與類�

( A ) 7. 下列程式片段執行後的輸出為何？ (A)20 10 (B)10 10 (c)10 20 (D)20 20。

struct ST{

int a; int b;

};

void F(struct ST \*p){ int tmp;

tmp = p->a; p->a = p->b; p->b = tmp;

}

int main() {

struct ST m, \*p; p = &m;

m.a = 10;

m.b = 20;

F(p);

printf("%d %d", m.a, m.b);

}



1.(B)

2.(D)

3.(B)

4.(B)

5.(D)

6.(c)

7.(A)

2. 「＆j是取址運算子。

1. pData 是指標，不可使用「.j運算子存取結構成員。
2. 結構指標變數 p 指向結構變數 box，選項(A)(B)(c)都會取得結構變數的成員 x 的值，選項(D)的 p 為結構指標變數，不可以使用「.j運算子存取指向的結構成員。
3. 程式執行的過程如下圖，
   1. 定義一結構 ST。
   2. 定義一函式 F(struct ST m)，參數 m 為結構變數，呼叫該函式時，傳入的引數必須也是結構變數。
   3. 宣告 ST 結構變數 m，使用「.j運算子存取結構成員。
   4. 呼叫 F(m)，屬於傳值呼叫，此時參數 m 可視為來源結構變數 m 的複製品(兩者皆為區域變數，允許同名)。
   5. 將參數 m 的成員 a 和 b 互換，不會改變來源結構變數 m 的成員。
   6. 函式結束，返回原程式位置。
   7. 結構變數 m 的成員 a 和 b 不變，輸出 10 和 20。

打 表示有詳解

程式設計實習

struct ST{ ·,-',

int a;

int b;

｝；

''. (l)

''. (2)

- void F(struct ST m){.---­

(4)

int tmp; tmp =m.a; m.a=m.b;

m.b =tmp;

·,

int main(){ struct ST m;

m.a =10; ·,

m.a=20;

''. (5)

''. (3)

(6)

* F(m);, \_. (7)

printf(11%d %d",m.a, m.b);.----

1. 程式執行的過程如下，

''. (l)

struct ST{.-­

int a;

int b;

｝；

* + void F(struct ST \*p){.----

1. int tmp;

''\_ (2)

tmp = p->a;

p->a = p->b; ｀

p->b = tmp;

'''(5)

int main(){

struct ST m, \*p; p=&m; ．，

m.a=lO; m.a=20;

-- (3)

(6)

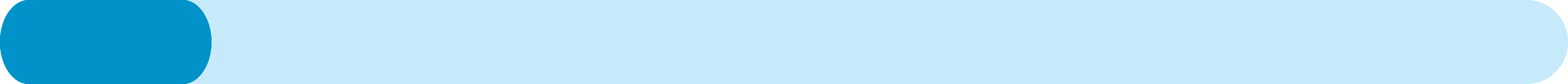
* + F(p);,,(7)

printf("%d %d",m.a, m.b); •

｝

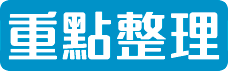
1. 定義一結構 ST。
2. 定義一函式 F(struct ST ＊p)，參數 p 為結構指標變數，呼叫該函式時，傳入的引數必須是結構變數的位址。
3. 宣告結構變數 m 和結構指標變數 p，使 p 指向 m。
4. 呼叫 F(p)，屬於傳址呼叫，此時參數 p 也會指向結構變數 m (參數 p 和結構指標變數 p 皆為區域變數，允許同名)。
5. 參數 p 使用指標運算子「->j將指向的結構變數成員 a 和 b 互換，會直接改變來源結構變數 m 的成員。
6. 函式結束，返回原程式位置。
7. 結構變數 m 的成員 a 和 b 已經被互換，輸出 20 和 10。

Chapter 7 結構與類�



7-4

類別



* + 1. c 語言是程序導向語言，c++是物件導向語言。
    2. 物件導向(Ob」ect-Oriented)語言，使用類別(class)定義資料的屬性與操作資料的方法，物件

(Ob」ect)則為類別的實作。

* + 1. c++語言宣告類別的語法(使用 **class** 關鍵字)，

class 類�名稱｛ private:

資料型態 資料成員；

回傳恒資料型態 成員函式（參數資料型態…）； public:

資料型態 資料成員；

回傳恒資料型態 成員函式（參數資料型態…）；

}；

類別的成員可以是資料或函式，

* 資料成員(data member)：即屬於該類別的變數，用來描述物件的屬性。
* 成員函式(member function)：即操作資料的方法，用來存取資料成員或進行運算。

而資料成員或成員函式所屬的區塊(private 或 public)，決定該成員的可存取範圍(權限)：

* **private** 區塊：成員只能在該類別內部被存取，此區塊內的成員稱為**私有成員**，可視為被封裝起來的隠藏性資料。
* **public** 區塊：成員可以在程式的任何位置被存取，此區塊內的成員稱為**公有成員**，一般做為存取資料成員的介面。

|  |
| --- |
| /\* 建立 date 類� \*/ |
| class date{ |
| /\* 私有成員區塊 \*/ |
| private: |
| /\* 資料成員 \*/ |
| int month; |
| int day; |
|  |
| /\* 成員函式，只宣告函式原型 \*/ |
| int getMonth(); |
| int getDay(); |
|  |
| /\* 公有成員區塊 \*/ |
| public: |

程式設計實習

|  |
| --- |
| /\* 資料成員 \*/ |
| int id; |
|  |
| /\* 成員函式，只宣告函式原型 \*/ |
| void setDate(int, int); |
| void printDate(); |
| }; |

1. 上例的 date 類別，只有宣告成員函式的函式原型，因此需在類別外部實作成員函式，語法如下，

回傳恒資料型態 類�名稱::成員函式（參數資料型態 參數名稱， …）｛ 程式碼區塊 }

|  |
| --- |
| int date::getMonth(){ |
| return month; |
| } |
|  |
| int date::getDay(){ |
| return day; |
| } |
|  |
| void date::setDate(int m, int d){ |
| month = m; |
| day = d; |
| } |
|  |
| void date::printDate(){ |
| cout << month << "/" << day << endl; |
| } |

「..j為範圍運算子。



註

1. 若在類別宣告時直接實作成員函式，則可以省略函式原型的宣告，

|  |
| --- |
| /\* 建立 date 類� \*/ |
| class date{ |
| /\* 私有成員區塊 \*/ |
| private: |
| /\* 資料成員 \*/ |
| int month; |
| int day; |
|  |
| /\* 不宣告函式原型，直接實作成員函式 \*/ |
| int getMonth(){ |
| return month; |

Chapter 7 結構與類�

|  |
| --- |
| } |
| int getDay(){ |
| return day; |
| } |
|  |
| /\* 公有成員區塊 \*/ |
| public: |
| int id; |
|  |
| /\* 不宣告函式原型，直接實作成員函式 \*/ |
| void setDate(int m, int d){ |
| month = m; |
| day = d; |
| } |
| void printDate(){ |
| cout << month << "/" << day << endl; |
| } |
| }; |

1. 完成類別宣告後，即可建立該類別的物件，宣告物件的語法如下，

類�名稱 物件名稱;

date birthday; /\* 建立birthday 物件，為date 類�的實作 \*/

1. 每個類別可以建立多個物件，這些物件皆為該類別的實作，稱為**實例(instances)**。
2. 物件可以直接存取公有成員，但私有成員無法直接存取，需透過公有成員函式，例如要存取 month 或 day 等私有資料成員， 需透過 setDate( )或 printDate( )。

birthday.printDate();

/\* 透過公有成員 setDate()，間接存取私有成員 month 和 day \*/

birthday.setDate(10, 10);

/\* 存取公有成員 id \*/

birthday.id = 1;

1. 「**this**j關鍵字，表示指向物件本身的指標，可以透個指標運算子「->j存取自身成員，但只能在類別內部使用，

|  |
| --- |
| void date::setDate(int m, int d){ |
| this->month = m; |
| this->day = d; |
| } |

1. 類別的成員函式可以**重載(overload)**，下例為 3 個重載的成員函式：

void setDate(int, int, int);

void setDate(double);

void setDate(int);

程式設計實習

1. **建構子(Constructor)**：或稱為建構函式，在宣告物件時，會自動呼叫建構子，建構子會對物件進行初始化。
2. 建構子的特性如下：
   1. 建構子需與類別同名。
   2. 建構子沒有回傳值，也不需加上 void 關鍵字。
   3. 一個類別可以擁有多個建構子，即建構子可以重載。
3. **解構子(Destructor)**：又稱為解構函式，當物件消滅時，會自動呼叫解構子，解構子執行物件結束時要進行的動作，如釋放記憶體或關閉檔案等。
4. 解構子的特性如下：
   1. 解構子需與類別同名，且名稱前面要加上「~j符號。
   2. 解構子沒有回傳值，也不需加上 void 關鍵字。
   3. 解構子只能有一個（不可重載），且不可傳入任何參數。
5. 下例中的 date 類別，有 2 個重載的建構子和 1 個解構子

|  |
| --- |
| /\* 建立 date 類� \*/ |
| class date{ |
| /\* 私有成員區塊 \*/ |
| private: |
| : |
|  |
| /\* 公有成員區塊 \*/ |
| public: |
| : |
| date(); /\* 建構子—的原型宣告 \*/ |
| date(int, int); /\* 建構子二的原型宣告 \*/ |
| ~date(); /\* 解構子的原型宣告 \*/ |
| : |
| }; |
|  |
| date::date(){ /\* 建構子—的實作 \*/ |
| month = 1; |
| day = 1; |
| } |
|  |
| date::date(int m, int d){ /\* 建構子二的實作 \*/ |
| month = m; |
| day = d; |
| } |
|  |

Chapter 7 結構與類�

|  |
| --- |
| date::~date(){ /\* 解構子的實作 \*/ |
| cout << "Goodbye" << endl; |
| } |
|  |
| date day1; /\* 宣告物件 day1，自動呼叫建構子— \*/ |
| date day2(12, 31); /\* 宣告物件 day2，自動呼叫建構子二 \*/ |

1. 類別的**繼承(inheritance)**：被繼承的類別稱為父類別，而繼承的類別稱為子類別，子類別會擁有父類別的公有成員，語法如下：

calss 子類� : public 父類�｛

:

}；

下例中的 policecar 類別繼承 car 類別，所以 policecar 類別會擁有 car 類別的成員函式

power( )，另外再新增一成員函式 flash( )，

|  |
| --- |
| /\* 宣告 car 類� \*/ |
| class car{ |
| private: |
| int id; |
| public: |
| void power(){cout << "啓動"}; |
| }; |
|  |
| /\* 宣告 policeCar 類�，繼承 car 類� \*/ |
| class policeCar : public car{ |
| public: |
| void flash(){cout << "閃燈"}; |
| }; |
|  |
| int main(){ |
| policeCar pCar; /\* 建立 pCar 物件 \*/ |
| pCar.flash(); /\* flash()是 policeCar 類�的成員函式 \*/ |
| pCar.power(); /\* power()是 policeCar 從 car 類�繼承的成員函式 \*/ |
| } |

使用「.j表示子類�繼承的父類�。



註

1. 子類別可以繼承多個父類別，子類別擁有所有父類別的公有成員，稱為多重繼承。

程式設計實習

|  |  |
| --- | --- |
| 私有成員 | 說明 |
| month | 月 |
| day | 日 |
|  |  |
| 公有成員 |  |
| setDate(int， int) | 設定日期 |
| printDate( ) | 顯示日期 |

***04 *** 

巧虎終於要有妹妹了，巧虎爸爸很高興小花妹妹要出 ，為此他打算改寫原來的程式，原程式使用結構來記錄巧虎的 日，參考範例 3，若改用 c++的物件導向功能來設計，就可以使用類別來定義資料的屬性和操作資料的方法，巧虎媽媽很支持爸爸的想法，主動幫忙規劃好 date類別，請幫忙他們完成新版的程式吧!(巧虎的 日是 5 月 5 日，小花的 日是 4 月 4 日) date 類別規劃：

程式碼：

|  |
| --- |
| 1 #include <iostream> |
| 2 using namespace std; |
| 3 |
| 4 class date{ |
| 5 private: |
| 6 int month; |
| 7 int day; |
| 8 |
| 9 public: |
| 10 void setDate(int, int); |
| 11 void printDate(); |
| 12 }; |
| 13 |
| 14 void date::setDate(int mm, int dd){ |
| 15 month = mm; |
| 16 day = dd; |
| 17 } |
| 18 |
| 19 void date::printDate(){ |
| 20 cout << "Your birthdat is " << month << "/" << day << endl; |
| 21 } |
| 22 |
| 23 int main() { |
| 24 date birthday1; |

Chapter 7 結構與類�

|  |
| --- |
| 25 date birthday2; |
| 26 |
| 27 birthday1.setDate(5, 5); |
| 28 birthday2.setDate(4, 4); |
| 29 |
| 30 birthday1.printDate(); |
| 31 birthday2.printDate(); |
| 32 } |
|  |

程式碼說明：

|  |  |
| --- | --- |
| 行數 | 說明 |
| 4 ~ 12 | 宣告 date 類別，成員如下：   * 2 個私有資料成員 month 和 day * 2 個公有成員函式 setDate( )和 printDate( ) |
| 14 ~ 17 | setDate( )函式實作。 |
| 19 ~ 21 | printDate( )函式實作。 |
| 24， 25 | 建立物件 birthday1 和 birthday2。 |
| 27 ~ 31 | 透過 setDate( )和 printDate( )成員函式，存取私有資料成員 month 和 day。 |

***05 *** 

酷巴克咖啡店的老闆利用賣咖啡的 閒時間，自修 c++程式語言，在學會物件導向的觀念後，他想將店內的咖啡產品物件化，並規劃一個 cafe 類別如下，請協助他完成程式後，產 兩杯咖啡物件，驗證程式執行結果。

cafe 類別規劃：

|  |  |
| --- | --- |
| 私有成員 | 說明 |
| price | 價格 |
| milk | 鮮奶量(％) |
| suger | 甜度(0 ~ 5 分) |
| setPrice(int) | 設定價格 |
| setMilk(int) | 設定鮮奶量(0 ~ 100) |
| setSuger(int) | 設定甜度(0 ~ 5) |
| int getPrice( ) | 取得價格 |
| int getMilk( ) | 取得鮮奶量 |
| int getSuger( ) | 取得甜度 |

程式設計實習

|  |  |
| --- | --- |
| 公有成員 |  |
| discount(float) | 設定折扣數 |
| show( ) | 顯示咖啡成份(奶量、甜度、價格) |
|  |  |
| 建構子 |  |
| cafe( ) | 預設建構子 |
| cafe(int int) | 建構子重載，初始化鮮奶量和甜度 |

程式碼：

|  |  |
| --- | --- |
| 1 | #include <iostream> |
| 2 | using namespace std; |
| 3 | |
| 4 | class cafe{ |
| 5 | private: |
| 6 | int price; |
| 7 | int milk; |
| 8 | int suger; |
| 9 | |
| 10 | void setPrice(int price) { |
| 11 | this->price = price; |
| 12 | } |
| 13 | void setMilk(int milk){ |
| 14 | this->milk = milk; |
| 15 | } |
| 16 | void setSuger(int suger){ |
| 17 | this->suger = suger; |
| 18 | } |
| 19 | |
| 20 | int getPrice(){ |
| 21 | return this->price; |
| 22 | } |
| 23 | int getMilk(){ |
| 24 | return this->milk; |
| 25 | } |
| 26 | int getSuger(){ |
| 27 | return this->suger; |
| 28 | } |
| 29 | |
| 30 | public: |

Chapter 7 結構與類�

|  |  |
| --- | --- |
| 31 | cafe(); /\* 建構子— \*/ |
| 32 | cafe(int, int); /\* 建構子二 \*/ |
| 33 | void discount(float); |
| 34 | void show(); |
| 35 | }; |
| 36 | |
| 37 | cafe::cafe(){ |
| 38 | setPrice(100); |
| 39 | setMilk(0); |
| 40 | setSuger(0); |
| 41 | } |
| 42 | |
| 43 | cafe::cafe(int m, int s){ |
| 44 | setPrice(100); |
| 45 | setMilk(m); |
| 46 | setSuger(s); |
| 47 | } |
| 48 | |
| 49 | void cafe::discount(float d){ /\* 依折扣重新計算價格 \*/ |
| 50 | setPrice(getPrice()\*d); |
| 51 | } |
| 52 | |
| 53 | void cafe::show(){ |
| 54 | cout << "奶量：" << getMilk() << "%" << endl; |
| 55 | cout << "甜度：" << getSuger() << "分糖" << endl; |
| 56 | cout << "價格：" << getPrice() << endl; |
| 57 | cout << "-----------" << endl; |
| 58 | } |
| 59 | |
| 60 | int main() { |
| 61 | cafe cup1; |
| 62 | cup1.show(); |
| 63 | |
| 64 | cafe cup2(40, 2); |
| 65 | cup2.discount(0.9); /\* 呼叫 discount 函式，打 9 折 \*/ |
| 66 | cup2.show(); |
| 67 | } |
| 68 | |

程式設計實習

程式碼說明：

|  |  |
| --- | --- |
| 行數 | 說明 |
| 4~35 | 依據類別規劃，宣告 cafe 類別，並直接實作私有成員函式，私有資料成員(price、milk、suger)均需透過私有成員函式(setPrice( )、setMilk( )、setSuger( ))存取。 |
| 37 ~ 47 | 實作 2 個重載建構子。 |
| 49 ~ 58 | 實作公有成員函式。 |
| 61 | 建立物件 cup1，自動呼叫建構子 cup( )。 |
| 64 | 建立物件 cup2，自動呼叫建構子 cup(int int)。 |



( B ) 1. 下列哪一種程式語言，不是物件導向語言？

(A)c++ (B)c (c)JAVA (D)Visual Basic。

( A ) 2. 關於 c++語言的類別建構子，下列哪一個說法是錯誤的？ (A)建構子的名稱和類別名稱不可以相同 (B)建構子支援重載

1. 建構子沒有回傳值 (D)一個類別可以有多個建構子。

( B ) 3. 關於 c++語言的類別，哪一個選項的描述是正確的？

* 1. 將資料成員放在 public 區塊，可以達到物件導向的資料封裝特性
  2. 建構子可以重載 (c)解構子可以重載

(D)建構子的名稱需與類別名稱相同，且需加上「~j符號。

( c ) 4. 在 c++語言中要建立類別，需使用哪一個關鍵字？ (A)define (B)array (c)class (D)struct。

( B ) 5. 在 c++語言中，要呼叫類別的成員函式，要使用哪一個運算子？ (A)「-j (B)「.j (c)「.j (D)「~j。

( D ) 6. c++語言允許函式重載，即函式的名稱相同，下列哪個選項內的函式，不是合法的重載函式？

1. void A(int)，

void A(float)，

(c) int A(float)，

float A(int)，

1. void A(int)，

void A(int， int)，

(D) void A(int)，

float A(int)，

Chapter 7 結構與類�

( A ) 7. 志遠宣告一個 c++的類別(class)如下，請問哪一個成員屬於私有資料成員，只能在物件內部存取？ (A)x (B)y (c)x 和 y 皆是 (D)x 和 y 皆不是。



class S{

private:

int x; public:

int y;

}

( D ) 8. 小麗使用 c++語言建立一個類別 star，下列哪一個函式不可能是該類別的建構子？ (A)star( )， (B)star(int)， (c)star(int， int)， (D)以上皆是。



( c ) 9. 下列關於建構子之敘述，何者為非？

(A)建構子也稱為建構函式，是物件的初始函式 (B)建構子支援重載(overload)

(c)建構子沒有傳回值，需加上 void (D)建構子與類別同名。

( B ) 10. 一程式片段如下，請問該程式執行後的輸出為何？

(A)HippoLion (B)MonkeyLion (c)BearLion (D)LionLion。

class Animal{ public:

~Animal(){

cout << "Lion";

}

Animal(){

cout << "Bear";

}

Animal(int x){

cout << "Monkey";

}

Animal(double x){ cout << "Hippo";

}

};

int main(void)

{

Animal a(1);

}

程式設計實習



1.(B)

2.(A)

3.(B)

4.(c)

5.(B)

6.(D)

7.(A)

8.(D)

9.(c) 10.(B)

3. (A)private 區塊 (c)解構子不可重載 (D)解構子。

1. 僅回傳值的型態不同，不是重載函式。
2. x 是私有成員，y 是公有成員。
3. (c)建構子沒有傳回值，也不需要加上 void。
4. 在 main()中宣告物件 a(1)時，會自動呼叫建構子 Animal (int x)，輸出"Monkey"，程式結束時會自動呼叫

解構子~Animal()，輸出"Lion"。

打 表示有詳解



( D ) 1. 關於 c++語言的結構(struct)和類別(class)，下列哪一個敘述正確？ (A)c++是物件導向語言，使用類別來定義資料和操作資料的方法 (B)結構是使用者自已建立的資料型態，包含多個成員

(c)一個類別可以有多個物件的實作 (D)以上皆是。

( B ) 2. 有關 c++語言的類別描述，下列何者錯誤？

(A)類別可以實現 c++物件導向的特性 (B)一個類別只能產 一個物件實例 (c)類別的建構子可以重載 (D)類別的解構子只能有一個。

( c ) 3. 有關 c++語言的物件導向功能，下列何者正確？

(A)c 語言繼承 c++語言，所以 c 語言也是物件導向語言

(B)使用「..j表示類別之間的繼承關係 (c)一個子類別可以繼承多個父類別 (D)類別就是物件的實作。

( c ) 4. 一程式片段如下，關於程式碼的說明，何者錯誤？

(A)I 類別繼承 J 類別 (B)I 類別會擁有 J 類別的公有成員

(c)J 類別擁有 beep( )成員函式 (D)I 和 J 是物件導向的繼承關係。

class I : public J{ public:

void beep();

( B ) 5. 小雪在 cIc++語言中宣告一個結構 box，程式碼如下，請問該結構會佔用多少的記憶體 間？ (A)3Bbyte (B)12Byte (c)24Byte (D)48Byte。



struct box{

int hight, length, width;

};

Chapter 7 結構與類�

( B ) 6. 承上題，使用 box 結構宣告 a 和 b 兩個結構變數，何者才是正確的語法？

1. a=struct(box)， b=struct(box)，

(c) box struct a， b，

1. struct box a， b，

(D) #struct box a

#struct box b

( A ) 7. 承上題，完成結構變數 a 和 b 的宣告後，執行下列程式片段的輸出為何？

(A)50 (B)40 (c)30 (D)20。

a.width = 50; b = a;

cout << b.width;

( c ) 8. 一程式片段如下，試問執行後的輸出為何？ (A)3 (B)4 (c)5 (D)6。

struct item{

int x, y;

};

int main()

{

struct item items[10];

for (int i=0; i<10; i++){ items[i].x = i / 2; items[i].y = i % 2;

}

cout << items[9].x + items[9].y << endl; return 0;

}

( c ) 9. 承上題，結構陣列 items[ ]會用掉幾 Bytes 的記憶體 間？

(A)40 (B)60 (c)80 (D)120。

( B ) 10. 函式 setValue( )的原型宣告如下，現有一 IOT 結構的結構變數 thing，下列何者是呼叫該函式的正確語法？

void setValue(struct IOT\*);

(A)setValue(thing)， (B)setValue(＆thing)，

(c)setValue(＊thing)， (D)setValue(IOT(thing))，。

程式設計實習

( A ) 11. 一程式片段如下，哪一個選項無法取得成員 x 的值？

(A)t->x (B)p->x (c)(＊p).x (D)t.xˤ

struct tree{ int x; int y;

};

int main(){

struct tree t; struct tree \*p; p = &t;

}

( A ) 12. 有關 c 語言的結構(structure)，下列的說明何者錯誤？

(A)結構用來定義資料和資料的操作行為 (B)一個結構可以包含 1 個以上的成員 (c)使用「.j運算子存取結構成員

(D)結構陣列的每個元素，都是相同的結構。 ( D ) 13. 下列有關 cIc++語言的敘述，何者是錯誤的？

(A)陣列是一個資料結構，可以存放多個相同資料型態的元素 (B)c++是物件導向語言，c 是程序導向語言

(c)c++支援函式重載(overload)，c 語言不支援

(D)自訂函式的回傳值型態為 null，表示函式沒有回傳值。

( D ) 14. 一程式片段如下，執行後的輸出為何？ (A)10 (B)20 (c)30 (D)60。

struct beta{

int x, y, z;

};

int main(){

struct beta a, b; struct beta \*p, \*q;

p = &a;

q = &b; (\*p).x = 10;

(\*p).y = 20;

(\*p).z = 30; q=p;

cout << q->x + q->y + q->z << endl;

}

Chapter 7 結構與類�

( c ) 15. 一程式片段如下，執行後的輸出為何？ (A)10 (B)50 (c)100 (D)編譯錯誤。

struct CAT{

int a, b;

};

void callCAT(struct CAT \*pCAT){ pCAT->a = 100;

pCAT->b = pCAT->a;

}

int main(){

struct CAT c; struct CAT \*p; p = &c;

p->a = 50;

callCAT(p);

cout << p->b << endl;

}

( D ) 16. 關於下列程式片段的描述，何者正確？

(A)x 是一個結構指標，指向結構變數 y (B)a 是結構變數 y 的其中一個成員

(c)y.a 的值是 1 (D)以上皆正確。

x = &y;

x->a = 1;

( c ) 17. c++語言的保留字有其特殊的含意，下列有關於保留字的說明，何者錯誤？ (A)break：中斷程式的執行，離開目前的程式區塊

(B)this：指向物件本身的指標 (c)const：建立自訂的結構型態

(D)continue：忽略後面的程式碼，立刻執行下一次迴圈。 ( c ) 18. 有關 c++語言的類別建構子，哪一個描述是錯誤的？

(A)建構子可以重載

(B)建構子的名稱必須和類別名稱相同 (c)建構子一定要有回傳值

(D)宣告物件時，會自動呼叫建構子，完成物件初始化。

( D ) 19. 有一函式原型 void key( )，下列何者不是其合法的重載函式？

(A)void key(int)， (B)int key(float)，

1. void key(int， int)， (D)int key( )，。

程式設計實習

( B ) 20. 關於下列程式碼片段的功能說明，哪一個是錯誤的？

* 1. 宣告一個 box 類別，有 4 個成員
  2. x 和 y 是公有成員，可以直接存取

(c)有 2 個重載的建構子，可以對物件初始化

(D)this 是指向物件本身的指標。

class box{

private:

int x, y; public:

box(){

this->x=10; this->y=10;

}

box(int x, int y){ this->x = x; this->y = y;

}

};

( c ) 21. 在 c++中宣告類別時，將資料成員宣告在哪個修飾子區塊內，可以達到物件導向的

「封裝j特性？ (A)public (B)hidden (c)private (D)close。

( A ) 22. 要將 c++中的成員函式做為對外的操作介面，達到資料隠藏的功能，必須使用哪一個修飾子來宣告成員函式？

(A)public (B)private (c)open (D)external。

( D ) 23. c++語言支援物件導向的功能，要在程式中宣告一個類別，需使用哪一個關鍵字？ (A)ob」ect (B)struct (c)define (D)class。

( B ) 24. 有關 cIc++語言的敘述，下列何者正確？

(A)c 和 c++語言都支援物件導向功能

(B)c 語言沒有字串(string)資料型別

(c)c++語言不支援函式重載

(D)c 是低階語言，c++是高階語言。

( c ) 25. c++語言的 this 關鍵字，表示指向物件本身的指標，要經由 this 存取自身的資料成員，需使用哪一個運算子？

(A)「.j (B)「.j (c)「->j (D)「..j。

( A ) 26. 小智宣告一個 c++的類別 pcb，下列哪個選項是其建構子的實作？

(A)pcb..pcb(int){ . }， (B)int pcb..layout( ){ . }，

1. void pcb..route( ){ . }， (D)void pcb..import(int) { . }，。

Chapter 7 結構與類�

( D ) 27. 執行下列程式片段後的輸出為何？ (A)1 (B)2 (c)3 (D)4。

class model{

private:

int val[5];

public:

void setNum(int, int); int getNum(int);

};

void model::setNum(int i, int num){ val[i] = num;

}

int model::getNum(int i){ return val[i];

}

int main(){

model m;

for (int i=0; i<5; i++){ m.setNum(i, i);

}

cout << m.getNum(4);

}

( D ) 28. 執行下列程式片段後的輸出為何？ (A)1 (B)2 (c)3 (D)4。

class base{

private:

int a, b;

void setA(int a){ this->a = a;

}

void setB(int b){ this->b = b;

}

public:

base(int, int);

void begin(int, int); int getA();

int getB();

};

base::base(int a, int b){ this->a = a;

this->b = b;

程式設計實習

}

void base::begin(int a, int b){ setA(b);

setB(a);

}

int base::getA(){ return this->a;

}

int base::getB(){ return this->b;

}

int main(){

base B(1, 2);

B.begin(3, 4);

cout << B.getA() << endl;

}



( B ) 1. 小華正在撰寫小論文的「引註資料j章節，因為參考的書籍太多，他想用 c++語言的結構來記錄書籍的相關資料，包含書名、作者、出版年份和 ISBN 書號等，下列關於結構的敘述，何者錯誤？

* 1. 自訂一結構資料型態，需使用 struct 關鍵字
  2. 結構的成員可以是整數、字串或函式

1. 宣告結構變數，可以同時指定成員的初始值 (D)結構變數使用「.j來存取成員。

( D ) 2. 接續上一題，小華使用下列程式碼建立一結構，請問關於這段程式碼的說明，何者錯誤？

* 1. 定義結構 book (B)該結構有 4 個成員 (c)有 2 個結構變數 (D)以上皆正確。

struct book{

int ISBN; int year; string name;

string author;

}myBook1, myBook2;

( c ) 3. 接續上一題，因為書籍資料太多，老師建議小華可以使用結構陣列的方式改寫程式，這樣就不用為每一本書宣告一結構變數，程式也會比較簡潔明暸，小華宣告一結構陣列 myBooks[20]，若要存取第 5 本書的 ISBN 資料，何者是正確的語法？

1. myBooks[4]->ISBN (B)myBooks[5]->ISBN

(c)myBooks[4].ISBN (D)myBooks[5].ISBN。

Chapter 7 結構與類�

( B ) 4. 小玲、小彤和小君三人組家參加物聯網創新應用大賽，她們的構想是發展智慧路燈，達到遠端控制路燈亮滅與監控的功能，小玲認為應該使用物件導向的程式設計方式，因為物件導向具有傳統結構化程式設計所沒有的特性，請問下列何者不是物件導向的特性之一？ (A)封裝 (B)雜湊 (c)多型 (D)繼承。

( D ) 5. 承上題，小彤使用 c++語言，建立一個對應眞實世界的路燈的類別，包含路燈的編號、經緯度座標、開關動作等，程式碼如下，請問哪個描述是錯誤的？

(A)類別名稱是 lamp (B)有 3 個成員是私有成員 (c)有 1 個不做任何動作的預設建構式 (D)只能用來宣告一個物件。 class lamp{

private:

int id;

int gps\_x, gps\_y; public: power\_on(); power\_off();

};

( A ) 6. 承上題，小君突然靈光一現，除了路燈之外，也可以將紅綠燈加入系統中，而且新增的紅綠燈類別，只要繼承原來的路燈類別，再加上屬於自已的成員即可，下列何者是正確的類別繼承語法？

(A)class trafficLight . public lamp { . }，

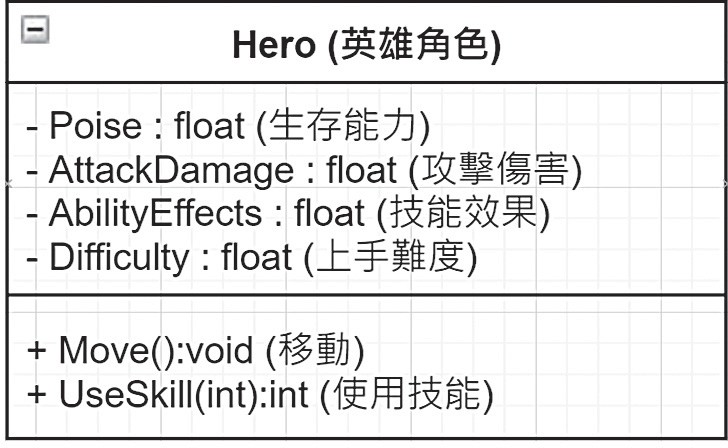
(B)class trafficLight .. public lamp { . }，

(c)class lamp . public trafficLight { . }， (D)class lamp .. public trafficLight { . }，。

題組：統一塑模語言(Unified Modeling Language，UML)是一種可以幫助開發物件導向程式的設計工具，在 UML 2.2 版本中定義了 14 種圖形，下圖是一個「類別圖j的範例，分為

3 個區域，由上而下分別是類別名稱、屬性(資料成員)和方法(成員函式)，而屬性和方法前面的「+j表示 public，「-j表示 private，此類別圖說明如下：

1. 類別名稱為 Hero。
2. 屬性皆為 float 型態。
3. 方法有 Move().void，沒有傳入參數和回傳值，UseSkill(int).int，參數和回傳值皆為 int 型態。



程式設計實習

( A ) 7. 請問若要使用 c++語言定義該類別，則下列程式碼中，哪一行不正確？ (A)001 (B)002 (c)007 (D)009。

|  |  |
| --- | --- |
| 001 | struct Hero{ |
| 002 | private: |
| 003 | float Poise; |
| 004 | float AttackDamage; |
| 005 | float AbilityEfficts; |
| 006 | float Difficulty; |
| 007 | public: |
| 008 | void Move(); |
| 009 | int UseSkill(int); |
| 010 | }; |

( B ) 8. 若要為該類別提供一建構子，以便在宣告物件時對物件進行初始化，請問下列何者是正確的建構子原型宣告語法？

(A)void Hero()， (B)Hero()， (c)void ~Hero()， (D)~Hero()，。

打表示有詳解

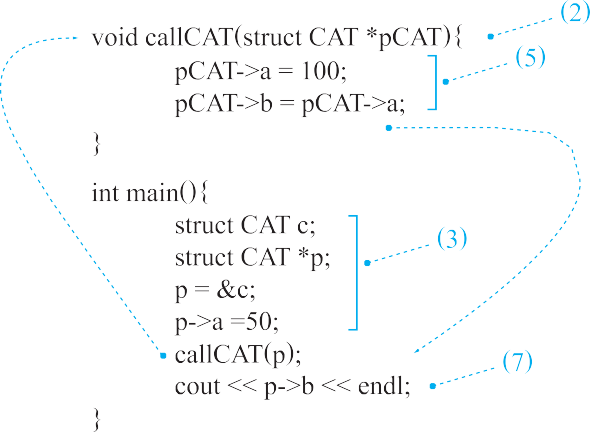
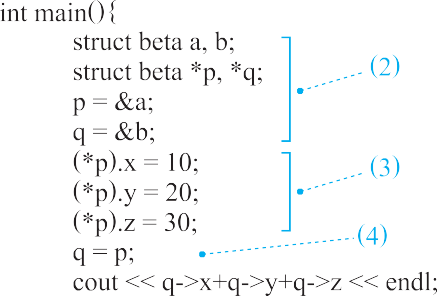
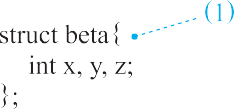


|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.(D) | 2.(B) | 3.(c) | 4.(c) | 5.(B) | 6.(B) | 7.(A) | 8.(c) | 9.(c) | 10.(B) |
| 11.(A) | 12.(A) | 13.(D) | 14.(D) | 15.(c) | 16.(D) | 17.(c) | 18.(c) | 19.(D) | 20.(B) |
| 21.(c)  ·素簣題  1.(B) | 22.(A)  2.(D) | 23.(D)  3.(c) | 24.(B)  4.(B) | 25.(c)  5.(D) | 26.(A)  6.(A) | 27.(D)  7.(A) | 28.(D)  8.(B) |  |  |
| I解斬I   1. (B)一個類別可以有多個物件的實作，稱為實例。 2. (A)c++繼承c 語言，c 不是物件導向程式。(B)使用「.j表示類別的繼承。(D)物件是類別的實作。 3. beep()是 I 類別的成員函式。 4. box 結構有 3 個 int 型態的成員，所需的記憶體大小為 3＊4 Byte=12 Byte。 5. 參考單元 7-2 立即練習第 5 題解析。 6. item 結構有 2 個 int 型態的成員，所需的記憶體大小為 2＊4 Byte = 8 Byte；結構陣列 items[]共有 10 個元素，每個元素都是 item 結構，故結構陣列 items[]佔用的記憶體大小為 10＊8 Byte = 80 Byte。 7. 函式原型 setValue(struct IOT＊)規範其參數為指向 IOT 結構變數的結構指標，因此呼叫該函式時，傳入的引數必須是 IOT 結構變數的位址。因題目未清楚說明 thing 的型態，故選項(A)和(B)皆正確，各選項的說明如下，    1. 若 thing 為 IOT 結構指標變數，且指向某結構變數，則選項(A)正確    2. 若 thing 為 IOT 結構變數，引數「＆thingj即為 IOT 結構變數的位址，則選項(B)正確   (c) 「＊j為取值運算子，故選項(c)一定不正確  (D) IOT 是結構不是函式，語法不正確   1. t 是結構變數，不可以使用指標運算子「->j存取結構的成員，故選項(D)的語法不正確。 2. c 語言限制結構的成員只能是變數，不能是函式，但 c++允許結構的成員是函式。 3. 沒有回傳值，需使用 void 型態。 | | | | | | | | | |



Chapter 7 結構與類�

1. 程式執行的過程如下：



* 1. 定義一結構 beta。
  2. 宣告結構變數 a 和 b、結構指標變數 p 和 q，使 p 指向 a，q 指向b。
  3. 存取 p 指向的結構變數成員 x、y、z (即結構變數 a 的成員)。
  4. 指定 q 等於 p，此時 q 和 p 都指向結構變數 a。
  5. 存取 q 指向的結構變數成員 x、y、z (即結構變數 a 的成員)。

1. 程式執行的過程如下：
   1. 定義一結構 cAT。
   2. 定義一函式 callcAT(struct cAT ＊pcAT)，參數 pcAT 是結構指標變數，呼叫該函式時，傳入的引數必須是結構變數的位址。
   3. 宣告結構變數 c 和結構指標變數 p，使 p 指向 c，指定結構變數 c 的成員 a 為 50。
   4. 呼叫 callcAT(p)，屬於傳址呼叫，此時參數 pcAT 也指向結構變數 c (參數 pcAT 和結構指標變數 p

都指向結構變數 c)。

* 1. 參數 pcAT 使用指標運算子「->j存取指向的結構變數成員，會改變來源結構變數 c 的成員。
  2. 函式結束，返回原程式位置。
  3. 結構變數 c 的成員 a 和 b 已經改變，輸出 100。

1. const 是宣告常數的關鍵字。
2. 建構子沒有回傳值，也不用 void 關鍵字。

程式設計實習

1. 重載函式的函式名稱必須相同，但參數個數或參數資料型態不同，選項(D)與題目的函式僅回傳值型態不同，不符合重載函式的條件。
2. 選項(B)x 和 y 是私有成員，必須透過公有成員函式存取。

24. (A)c 語言不支援物件導向 (c)c 語言不支援函式重載，c++支援函式重載 (D)c 和 c++都是高階語言。

1. 建構子需與類別同名，且沒有回傳值(不需加上 void)。
2. 程式執行的過程如下，

·· ·· (l) class model{..---

private:

int val[5];

public:

void setNum(int, int); int getNum(int);

｝，

void rr\_i�del::setNum(int i, int num){ val[i] = num;

(2)

•·

｝

int model::getNum(int i){

return val[i];

｝

int mainO{ (3)

model m;.---

for (int i=O; i<5; i++){

m.setNum(i, i);

.-\_.--- (4)

cont<< m.getNum(4);.. ,.- (5)

* 1. 宣告類別 model，其中私有資料成員為陣列 val[]，公有成員函式只宣告函式原型。
  2. 實作成員函式 setNum()和 getNum()。
  3. 建立類別 model 的物件 m。
  4. 呼叫公有成員函式 setNum()，指定私有資料成員 val[]的元素值。
  5. 呼叫公有成員函式 getNum()，取得私有資料成員 val[4]的值，輸出 4。

Chapter 7 結構與類�

1. 程式執行的過程如下，

class base{ ·',

private:

int a, b;

,,.(l)

void setA(int a){

this->a = a;

｝

void setB(int b){ ·,

this->b = b;

,,(2)

｝

public:

base(mt, mt); (3) void begin(int, int);.

int getAO;

int getB();

(6)

｝；

* base::base(int a, int b){

this->a = a;

．，

this->b = b;

｝

\_..(4)

(7)

* + void base::begin(int a,int b){ setA(b);

setB(a);

｝

int base::getA(){ return this->a;

｝

int base::getBO{ return this->b;

｝

int main(){

* + - base B(l, 2);
    - B begin(3, 4); cout<<B.getAO<<endl;.---

｝

．，

,,-(8)

''. (5)

1. 宣告類別 base。
2. 類別宣告時，直接實作私有成員函式。
3. 類別宣告時，公有成員函式只有宣告函式原型。
4. 實作建構子 base()。
5. 實作公有成員函式 begin()、getA()和 getB()。
6. 建立類別 base 的物件 B，會自動呼叫建構子 base(int a， int b)，指定私有資料成員 a 和 b 的值。
7. 呼叫公有成員函式 begin()，在 begin()內會再呼叫私有成員函式 setA()和 setB()，指定私有資料成員 a 和 b 的值。
8. 呼叫公有成員函式 getA()，回傳私有資料成員 a。

程式設計實習



( C ) 1. 關於 C 語言中結構(structures)資料型態的敘述，下列何者錯誤？

* 1. 結構是一種自行定義的資料型態
  2. 宣告結構資料型態的變數時，只能使用已定義的結構名稱宣告 (C)定義結構只能使用不同資料型態的變數

1. 定義結構只能使用 struct 關鍵字。 【111 年統測】

( A ) 2. 小文設計了一個計算矩形面積的類別 CalculateArea，其宣告如下，則下列哪一個成員函式(Member Function)定義正確？

* 1. class CalculateArea{
  2. private:
  3. double Length, Width, Area;
  4. public:
  5. void SetPara(double, double);
  6. double GetLength();
  7. double GetWidth();
  8. double GetArea();

9 };

(A)void CalculateArea:: SetPara(double L, double W){Length = L;Width = W;}

(B)double GetLength( ){return Length;} (C)CalculateArea:: GetWidth( ){return Width;}

(D)double CalculateArea:: GetArea( ){ Area = Length \* Width;}。 【112 年統測】

( C ) 3. 在沒有使用存取修飾字(private、public 或 protected)的情況下，下列敘述何者正確？

(A)class 內的所有成員變數都可在 main ( )主程式內存取

(B)struct 宣告的物件不可以做為 class 的成員

(C)任何函式內的 struct 型態變數中的成員，在該函式的範圍內都可以被存取

(D)class 內的成員函式不可存取同一個 class 內的成員變數。 【112 年統測】

A閱讀下文，回答第 4~5 題

C 程式語言中的 typedef 關鍵字，可以將複雜的資料型態用簡單的別名來取代。例如下列資料型態宣告與程式片段，以 id 來儲存一個學 的學號，score 來儲存該學 的成績。假設班上共 N 個學 ，第 i 個學 的資料儲存在 student[i–1]裡面。

1. #include <stdio.h>
2. #define N 50
3. void main(){
4. typedef struct studentScore {
5. int id; //學號
6. float score; //成績
7. } SCORE;
8. SCORE student[N], \*p;
9. float sscore;
10. p = student+28;
11. ...

Chapter 7 結構與類�

( A ) 4. 在行號 10 之後，若要取得學號 28 的學 成績放到變數 sscore，下列程式碼何者正確？ 【113 年統測】

(A) sscore = student[27].score; (B) sscore = SCORE[27].student;

(C) sscore = student – > score[27]; (D) sscore = student – > score[29];。

( B ) 5. 使用泡沫排序演算法來將 student 陣列中的成績 score 排序時，關於此演算法需經幾次的成績數值比較，可得排序結果？

(A)50 (B)1225 (C)24550 (D)245050。 【113 年統測】

A閱讀下文，回答第 6~7 題

下列程式片段為計算管道內水流量的全域類別，其中 TotalFlow()為計算流量值的成員函式。

1. enum Item{\_FlowRate, \_Time };
2. class Volume {
3. private:
4. double FlowRate, Time;
5. static double Offset;
6. public:
7. Volume (){ FlowRate =0.0, Time = 0.0;}
8. Volume (double In1, double In2)
9. { FlowRate = In1; Time = In2; }
10. void SetOffset(double offs) { Offset = offs; }
11. double GetParameter(Item item)
12. { return (item == \_Time) ? Time : FlowRate ;}
13. void SetPara(double FR,double T)
14. { FlowRate =FR; Time = T;}
15. protected:
16. double TotalFlow(){return FlowRate\*Time;}
17. } T1(1.0,2.3), T2;
18. double Volume :: Offset = 0.0;

( D ) 6. 若要在 main 主程式內使用 T1 或 T2 物件來編寫程式，下列程式敘述何者正確？

* 1. double Value1 = T1 – > GetParameter (\_Time);
  2. T1 – > SetPara (100.5,50);
  3. T2.FlowRate = 12.4;
  4. T2.SetOffset (–3.2);。 【113 年統測】

( A ) 7. 根據 Volume 類別的宣告，程式碼都在 main 內執行，下列敘述何者正確？

1. 執行 T2.SetOffset(0.5); 則 T1 內的 Offset 也會被同時修改為 0.5
2. T1 與 T2 內的 FlowRate 預設值都為 0.0
3. 執行 double X = T1.TotalFlow(); 可以將流量值回傳給 X
4. 執行 float Y = T2.GetParameter((Item)1); 可以將 FlowRate 值回傳給 Y。

【113 年統測】

程式設計實習

打 表示有詳解



1.(C) 2.(A) 3.(C) 4.(A) 5.(B) 6.(D) 7.(A)

I解斬I

1. (C)結構的成員可以是相同的資料型態。
2. 在類別外部實作成員函式的語法如下，

**回傳值資料型態 類別名稱::成員函式(參數資料型態 參數名稱, …){ 程式碼區塊 }**

故選項(B)和(C)的語法錯誤，正確程式碼應為

(B) double **CalculateArea**:: GetLength( ){return Length;}

(C) **double** CalculateArea:: GetWidth( ){return Width;}

選項(D)未將 Area 回傳，屬於執行階段錯誤，正確寫法應為

(D) double CalculateArea:: GetArea( ){ **return** Area = Length \* Width;}

1. 未指定修飾子的情況下，結構（structure）的成員預設皆為 public，類別（class）的成員預設皆為 private
   1. class 的資料成員只能在內部存取
   2. class 的資料成員可以是基本資料型態（int、double、…），也可以是複合資料型態（array、struct）

(D) 類別內的成員函式，可以存取同一類別內的資料成員

1. 參考單元 7-2 立即練習第 2 題。
2. 參考單元 5-1 範例 4，*n* 筆資料進行泡沫排序演算法，需比較  *n*(*n* 1) 次。

2

1. 指標運算子「– >j需配合「thisj關鍵字使用，選項(A)(B)的正確語法如下，
   1. double Value1 = T1.GetParameter (\_Time);
   2. T1.SetPara (100.5, 50);
   3. FlowRate 為 private 成員，不可直接存取。
2. (A)宣告區域變數時若加上「staticj關鍵字，則該變數會成為靜態變數(static variable)，靜態變數每一次宣告皆會配置到相同的記憶體位置，且其值會維持到程式結束，不會因為再次宣告而改變。如下列程式

碼，x 為靜態變數，雖每次呼叫 foo()都會宣告 x，但其值會一直維持到程式結束，程式執行後輸出 123。 void foo(){

static int x=0; //宣告 x 為靜態變數 x++;

std::cout << x;

}

int main(){

foo();

foo();

foo();

}

而類別中的靜態資料成員，是由所有該類別的實例共用，例如此題中 Offset 為 Volume 類別的靜態資料成員，T1、T2 為 Volume 類別的實例，則 T1 和 T2 會共用 Offset。

(B)宣告 T1 時，會呼叫建構子 Volume(double In1, double In2)，故 T1 的 FlowRate 初始值為 1.0、Time

初始值為 2.3。

Chapter 7 結構與類�

1. protected 修飾子與 private 修飾子相似，成員只允許在內部存取，但若考慮繼承關係時，則子類別可以存取父類別的 protected 成員，卻無法存取父類別的 private 成員。ToatlFlow()屬於 protected 成員，所以無法直接存取。
2. 傳入的引數(Item)1 為 Item 列舉的第 2 個成員\_Time，GetParameter()會回傳 Time 的值。

程式設計實習