

# 《程式語言》

一、請宣告一個Add class，內容包含以下特性：（25分）

- Private member "A"，型態為integer，並透過constructor初始化為0。
- Private member "B"，型態為integer，並透過constructor初始化為0。
- Private method "print"，將member "A"、"B"相加後顯示在console中，且無回傳值。
- Public method "add(int A, int B)"，須達成以下行為：
  - 將參數A的value儲存到member "A"。
  - 將參數B的value儲存到member "B"。
  - 呼叫Private method "print"，印出結果。
  - 無回傳值。

```
public class Add {
    // code here
}
```

試題評析	本題為Java類別相關考題，相較於近年相關試題，題型結構單純，配分也不少，是考生應完全掌握的題目。值得注意的是，題目中對於資訊隱藏層級、變數型態及回傳值都有加以規定，不能憑經驗自行修改，如成員屬性的值必須要初始化，且必須用資訊隱藏保護為private變數；而成員方法亦為private保護，若考試時心浮氣躁，沒注意到細節，就有可能因作答不合題意被扣分，千萬要小心謹慎。另外add()方法因為形式參數與數行名稱相同，必須要透過this來指定，亦要特別注意。
考點命中	《高點·高上程式語言講義》第四回，金乃傑編撰，頁20-21。

答：

以Java撰寫程式如下：

```
3 class Add{
4     private int A;           //私有成員屬性，整數A
5     private int B;           //私有成員屬性，整數B
6
7     private void print(){
8         int sum = A + B;     //將A與B相加
9         System.out.print(sum); //顯示在console中
10    }
11
12    Add(){                    //建構子初始化A與B
13        A = B = 0;
14    }
15
16    public void add(int A, int B){
17        this.A = A;           //更新屬性A
18        this.B = B;           //更新屬性B
19        print();              //印出結果
20    }
21 }
22
23 public class K106_1 {
24     public static void main(String[] args) {
25         Add a1 = new Add();
26         a1.add(3, 5);
27     }
28 }
```

註：本題第23-28行考試時可不用寫，並建議依照題意將第3行類別宣告為public class A會較為正確。

二、程式語言中，指標 (Pointers) 是一個非常強大有用的工具，但是也可能為程式帶來Dangling Pointer和Memory Leakage問題。(25分)

(一)請使用下列的C程式碼說明何謂Dangling Pointer和Memory Leakage問題，並解釋其發生原因。

```
int *p1, *p2, *p3, *p4;
p1 = (int*) malloc (sizeof(int));
p2 = p1;
free p1;
p3 = (int*) malloc (sizeof(int));
p4 = (int*) malloc (sizeof(int));
p3 = NULL;
```

(二)請從Compiler的角度，說明什麼方法可以解決這兩個問題？

試題評析	本題使用較久沒出現的考點—指標與懸置問題，先要求學生做名詞解釋，再說明程式中造成此現象的原因，最後要求說明解釋方法。考題相似於100年鐵路三等之考題，並無太大變化，考生只要掌握好題目中此二現象之造成原因及解決辦法即可獲得不錯分數。
考點命中	《高點·高上程式語言講義》第三回，金乃傑編撰，頁34-36。

答：

(一)以下依照題目中程式碼說明Dangling Pointer及Memory Leakage問題及原因。

1.Dangling Pointer：指懸置指標，意即指標所指到的物件以不具意義的情況，如題目中p2之指標。以下抄寫題目之片段程式說明：

程式碼	說明
p1 = (int *)malloc(sizeof(int));	p1指向一個以malloc產生之整數物件。
p2 = p1;	p2指向與p1相同之物件(p2複製p1中所存放的記憶體位置)。
free p1;	手動釋放p1所指向之整數物件。 由於p2也指向該物件之記憶體位址，但該物件卻被釋放，因此使p2成為懸置指標(dangling pointer)。

2.Memory Leakage：指懸置物件，意即一物件無任何指標指向之情形。由於沒有指標指向，因此程式無法操作該物件，但該物件卻佔用記憶體空間，因此稱為記憶體垃圾，如題目中p3所指向的整數空間。以下抄寫題目之片段程式說明：

程式碼	說明
p3 = (int *)malloc(sizeof(int));	p3指向一個以malloc產生之整數物件。
p4 = (int *)malloc(sizeof(int));	p4指向一個以malloc產生之整數物件。
p3 = NULL;	重新指派指標p3，使其指向NULL。 由於p3指向NULL，故原被p3指向之整數物件便沒有任何一個指標指向，故稱之為記憶體洩漏(Memory Leakage)。

(二)從Compiler的角度，可以在建立指標或物件加上特殊處理，方法說明如下：

1.懸置指標 (Dangling Pointer) 可使用墓碑法 (tombstones)，做法如下：

- (1)每塊記憶體各有一個墓碑，指向該記憶體的指標其實指向它的墓碑。
- (2)若記憶體被釋放，墓碑值設定為null。
- (3)當指標要存取記憶體時，發現墓碑值為null則不允許存取，如此可以確保指標指向已經被釋放的物件。

2.記憶體洩漏 (Memory Leakage) 可使用參考計數法 (reference-counting)，作法說明如下：

- (1)每個物件增加一個計數(counter)欄位。
- (2)若有被參考到則計數欄位值加一，取消參考時則減一。
- (3)若計數欄位值為0，則回收該物件記憶體，如此可以確保沒有指標參考到的物件都能自己釋放掉。

以上兩種方法都是透過編譯器在轉換程式碼時增加特定欄位，來確保記憶體能被有效利用，實際上相關的方法很多，也可以交互使用。但若配合程式設計師在開發時以較嚴謹的方式撰寫，則才是根治此問題並且提高效率的解決之道。

三、下列是以C，Java，Javascript，C++，PHP及JSP六種語言撰寫存取陣列（array）的程式片斷，請問此程式片斷執行後，會產生什麼結果？如果有結果錯誤，錯誤訊息為何？各用什麼方法可避免錯誤？請寫出完整的程式來說明你的方法。（25分）

C: `int a[] = {0, 1, 2, 3}; for(i=0; i<=sizeof(a); i++) printf("%d", a[i]);`

Java: `int[] a = {0, 1, 2, 3}; for(i=0; i<=a.length; i++) System.out.print(a[i]);`

Javascript: `var a = [0, 1, 2, 3]; for(i=0; i<=a.length; i++) document.write(a[i]);`

C++: `int a[] = {0, 1, 2, 3}; for(i=0; i<=sizeof(a); i++) printf("%d", a[i]);`

PHP: `$a = array(0,1,2,3); for($i=0; $i<=count($a); $i++) echo $a[$i];`

JSP: `int[] a = {0, 1, 2, 3}; for(i=0; i<=a.length; i++) out.print(a[i]);`

<b>試題評析</b>	本題相似於99年地方特考三等，但使用六種程式語言，透過for迴圈中敘述的改動，造成陣列發生所引超出範圍。並要求考生說明原因及解決方法。值得注意的是，題目中要求要寫出完整程式，但是題目中給的程式卻是片段的，在某些要求較嚴謹的語言缺乏主類別或主程式，並沒有宣告整數變數i，因此完全無法執行。推測題意僅是因大部分錯誤點在迴圈測試運算式中，因較難描述才要求考生寫出完整程式說明，因此考慮在作答前先加上說明，避免因程式不構完整而被扣分。另本題需要對C/C++中的sizeof()及PHP中的count()運作結果較熟悉，才能知道修改方向。
<b>考點命中</b>	《高點·高上程式語言講義》第三回，金乃傑編撰，頁111。

**答：**

假設在C/C++、Java版本中均有主要類別或主程式，且變數i有事先宣告，以下將題目中六種程式語言執行的結果(或錯誤)，及解決方法與完整程式碼以表格整理說明如下：

程式	執行結果或錯誤	解決方法與程式碼
C	在UNIX系統會當機，而在Windows系統執行結果如下： 0123後面再印出一大段隨機數字 說明：該大段隨機數字實際上為在a[]後方的為初始化記憶體中之內容，不具意義。	問題來自for迴圈中測試運算式，因sizeof計算的是整個陣列所佔byte數，因此必須再除以sizeof(int)才能計算陣列元素數，此外必須使用小於(<)而非小於等於(<=)，不然還是會超出陣列範圍。修改如下： <code>int a[] = { 0, 1, 2, 3 }; for (i = 0; i &lt; sizeof(a)/sizeof(int); i++) printf("%d", a[i]);</code>
Java	先印出0123再跳出例外程式結束。產生之例外如下： <code>java.lang.ArrayIndexOutOfBoundsException</code> 說明：例外來自於要印出未定義的a[4]	問題來自for迴圈中測試運算式，使用length可取出陣列長度，但在Java中陣列注標是從0開始，因此使用小於等於(<=)會使迴圈走到超出陣列注標。修改如下： <code>int[] a = { 0, 1, 2, 3 }; for (i = 0; i &lt; a.length; i++) System.out.print(a[i]);</code>
JavaScript	0123undefined 說明：JavaScript遇到要輸出未定義的a[4]時，會印出undefined的字樣。	問題來自for迴圈中測試運算式，使用length可取出陣列長度，但在JavaScript中陣列注標是從0開始，因此使用小於等於(<=)會使迴圈走到超出陣列注標。修改如下： <code>var a = [ 0, 1, 2, 3 ]; for (i = 0; i &lt; a.length; i++) document.write(a[i]);</code>

程式	執行結果或錯誤	解決方法與程式碼
C++	0123後面再印出一大段隨機數字 說明：問題同C語言版本	將程式修改如下： <pre>int a[] = { 0, 1, 2, 3 }; for (i = 0; i &lt; sizeof(a)/sizeof(int); i++)     printf("%d", a[i]);</pre>
PHP	0123 說明：在某些設定較嚴謹的PHP直譯器會在上述輸出後跳出NOTICE警告，因要輸出未定義的\$a[4]；但若關閉NOTICE警告，PHP可正確執行。	問題來自for迴圈中測試運算式，使用count可計算陣列中元素數，但在PHP中陣列注標是從0開始，因此使用小於等於(<=)會使迴圈走到超出陣列注標。修改如下： <pre>\$a = array( 0, 1, 2, 3 ); for (\$i = 0; \$i &lt; count(\$a); \$i++)     echo \$a[\$i];</pre>
JSP	先印出0123再跳出例外程式結束。產生之例外如下： java.lang.ArrayIndexOutOfBoundsException 說明：例外來自於要印出未定義的a[4]	問題來自for迴圈中測試運算式，使用length可取出陣列長度，但在JSP中陣列注標是從0開始，因此使用小於等於(<=)會使迴圈走到超出陣列注標。修改如下： <pre>int[] a = { 0, 1, 2, 3 }; for (i = 0; i &lt; a.length; i++)     out.print(a[i]);</pre>

四、下列為秀出4x3表格的html網頁，請以此網頁為基礎，用Javascript程式完成所需的兩個js檔案：data.js及function.js。其中function.js只包括函式ShowTable(I, J)及data.js只包括所需的二維陣列宣告。一定要用Javascript的for迴圈完成函式ShowTable(I, J)。(25分)



```
<HTML><HEAD><TITLE>4x3 Table Example</TITLE>
<SCRIPT SRC=Data.js</SCRIPT>
<SCRIPT SRC=Function.js</SCRIPT>
</HEAD><BODY>
<SCRIPT LANGUAGE=JavaScript> ShowTable(4,3); </Script>
</BODY></HTML>
```

<b>試題評析</b>	本題為HTML及JavaScript考題，除了要熟悉迴圈的運作方式，亦需要掌握HTML的表格撰寫方法及JavaScript的二維陣列寫法。由於JS中變數都不需要事先宣告，因此二維陣列的寫法就有好幾種。一般來說，考生只要使用自己熟悉的寫法撰寫及可，若擔心寫錯也可以將陣列中的內容分開指定。值得一提得是，觀察題目中網頁表頭有加上Key1~3，估計應不是陣列內資料（因題目中說表格是4×3），故透過迴圈中針對key去分別產生處理。由於題目難度不高，僅在部分語法細節不易掌握，若能掌握，得分應能符合預期。
<b>考點命中</b>	《高點·高上程式語言講義》第五回，金乃傑編撰，頁21-24、31。

**答：**

以JavaScript撰寫如下：

Function.js檔案內容：

```
function ShowTable(I,J){
    document.write("<table border=1>");
    for(var i = 0; i <= I; i++){
        document.write("<tr>");
        for(var j = 0; j < J; j++){
            if(i == 0) document.write("<td>Key "+(j+1)+"</td>");
            else document.write("<td>"+data[j][i-1]+"</td>");
        }
        document.write("</tr>");
    }
    document.write("</table>");
}
```

Data.js檔案內容：

```
var data = [];
data[0] = [3, 4, 7, 10];
data[1] = [2, 5, 8, 11];
data[2] = [1, 6, 9, 12];
```

輸出結果：

Key 1	Key 2	Key 3
3	2	1
4	5	6
7	8	9
10	11	12

【版權所有，重製必究！】