

《程式語言》

試題評析	<p>今年程式語言的考題算是簡單。第一題比較命令式語言與函數型語言在變數使用上之優缺點；第二題是文法推導的問題；第三題考傳值呼叫與傳位址呼叫這兩種參數傳遞方法；第四題考捷徑計算；第五題考 C++ 的虛擬函數與非虛擬函數之差異。整體而言都不困難，只要仔細寫應可得到不錯的成績。</p> <p>今年的考題對普通的考生而言，約可得 60 分，程度好者可望拿 80 分以上成績。</p>
------	---

一、如 C 等之程序語言 (procedural programming language) 和如 Common Lisp 等之函數型語言 (functional programming language) 主要差別是變數 (variable) 及指定指令 (assignment) 之使用，試說明其優缺點。(二十分)

【擬答】

- (一) 在 C 等命令式語言中，程式的執行主要是透過變數 (variable) 及變數值的指定 (assignment) 來完成計算工作，亦即以變數儲存資料並由程式師自行管理，其優點包括：
1. 提供很大的使用彈性，利用變數可以設計各種重複結構；
 2. 透過變數管理記憶體資料，可提供較高的執行效率。
- 缺點如下：
1. 由程式師自行管理變數儲存資料，是一種較低階的程式設計方法。
 2. 容易產生副作用 (side effect)，尤其函數中的副作用可能會降低程式之可靠度。
- (二) 在 Common LISP 等函數型語言中，程式的執行主要是透過函數及函數之應用來完成工作，可以不使用變數及指定敘述，其優點包括：
1. 只需將資料經由參數傳給函數作運算，不用去管理變數來儲存資料，是一種較高階的程式設計方法；
 2. 因為函數中不使用變數，故不會有副作用的問題，只要輸入的參數相同，即可得到相同的函數值。
- 缺點如下：
1. 因不使用變數，故無法設計各種反覆迴圈，迴圈是經由函數的遞迴呼叫和條件運算式所構成。
 2. 程式執行的效率較低。

二、下述文法會產生二進位數字，試說明其產生數字之值為那一數字之倍數。(二十分)

$\langle C \rangle \rightarrow \langle C \rangle 0 \mid \langle A \rangle 1 \mid 0$

$\langle A \rangle \rightarrow \langle B \rangle 0 \mid \langle C \rangle 1 \mid 1$

$\langle B \rangle \rightarrow \langle A \rangle 0 \mid \langle B \rangle 1$

【擬答】

此文法所產生的數字皆為 3 的倍數。例如

$\langle C \rangle \Rightarrow \langle A \rangle 1 \Rightarrow 11 \dots\dots\dots 3$

$\langle C \rangle \Rightarrow \langle C \rangle 0 \Rightarrow \langle A \rangle 10 \Rightarrow 110 \dots\dots\dots 6$

$\langle C \rangle \Rightarrow \langle A \rangle 1 \Rightarrow \langle B \rangle 01 \Rightarrow \langle A \rangle 001 \Rightarrow 1001 \dots\dots\dots 9$

以此類推。



三、考慮下列程式片段：

```
main() {  
    int value = 3; list[5] = {2, 4, 6, 8, 10};  
    swap (value, list[0]);  
    swap (list[1], list[3]);  
    swap (valne, list[value]);  
}  
  
void swap (int x, int y){  
    int temp;  
    temp =a;  
    a = b;  
    b = temp;  
}
```

- (一)若參數傳遞為以值傳遞 (passed by value)，則每次 (共 3 次) swap 執行後 list 及 value 之結果為何？(十分)
- (二)若參數傳遞為以址傳遞 (passed by address)，則每次 (共 3 次) swap 執行後 list 及 value 之結果為何？(十分)

【擬答】

- (一)參數以 pass by value 傳遞，每次執行後實際參數值皆不受影響，即結果為
value = 3, list[5] = {2, 4, 6, 8, 10}
- (二)參數以 pass by address 傳遞，每次執行時實際參數值皆會受影響，結果為
value = 6, list[5] = {3, 8, 2, 4, 10}

四、(一)何謂捷徑計算 (short-circuit evaluation)？(十分)

(二)試說明當一種程式語言不提供捷徑計算時，會發生什麼問題，並舉例說明之。(十分)

【擬答】

- (一)捷徑計算是指運算式的結果毋須計算所有的運算子或運算元即可決定的計算方式。
- (二)當一個程式語言未提供捷徑計算時，程式的可靠度會較低。例如 PASCAL 沒有提供捷徑計算能力，若要在一個長度為 LENGTH 的陣列中尋找一個值與 VALUE 相同的元素，則下列 PASCAL 程式：

```
INDEX := 1;  
while (INDEX <= LENGTH) and (LIST[INDEX] <> VALUE) do  
    INDEX := INDEX + 1;
```

有可能發生陣列註標超出範圍(index out of range)的錯誤，因為當 INDEX 大於 LENGTH 時 (表示找不到)，卻仍需要去存取 LIST[INDEX]，此時即會發生此種錯誤。



五、考慮下列 C++ 程式片段：

```
class B {  
    public:  
    virtual char f() {return 'B';}  
    char g() {return 'B';}  
    char testF() {return f();}  
    char testG() {return g();}  
}  
class D: public B {  
    char f() {return 'D';}  
    char g() {return 'D';}  
}  
main() {  
    D d;  
    print d.testF(), d.testG();  
}
```

則執行 d.testF() 及 d.testG() 之結果為何？並請說明之。(二十分)

【擬答】

執行之後輸出如下：

DB

因為類別 B 中的 f() 宣告為虛擬函數(virtual function)，可支援多模(polymorphism)之功能；而函數 g() 僅為一般的函數成員，並未支援多模。故 d.testF() 執行時，會依照接收者物件為 d，而執行其所屬類別 D 中定義的 f()，故輸出 D；而 d.testG() 執行時，則直接執行 d.testG() 所屬類別中的函數成員 g()，而與接收者物件是誰無關，故輸出 B。

