

《程式語言》

試題評析	今年程式語言的考題都算是很基本的題目，答案也都不複雜，本班的學生應該都可以拿到不錯的分數。前兩題都是基本的觀念性問題，其中第一題考例外處理的意義並舉例說明其功能，第二題考資料抽象化的意義與舉例說明其在程式語言中的功能，這兩題只要觀念正確應不難作答；第三題考河內塔的遞迴程式設計問題，程式本身不難，但必須清楚了解河內塔問題的遞迴式解法，才能正確地將程式寫出來；第四題考兩段程式指令的時間複雜度，程式很簡單，算是送分題；最後一題考文法語句之推導，也是很簡單的問題。綜合而言，今年的題目普通的考生應可拿 60~70 分左右，程度較好者 80~90 分沒有問題。
------	---

一、(一)何謂例外處理 (exception handling) ? (10 分)

(二)例外處理在程式語言之功能為何? 舉例說明之。(10 分)

【擬答】

(一)例外處理是指當系統偵測到由軟體或硬體所引發的例外時，會主動執行一個稱為例外處理程式 (exception handler) 之程式單元來處理此一例外狀況。

(二)例外處理在程式語言中可將錯誤 (例外) 處理指令與一般指令分離，讓程式處理流程較為清晰單純，以簡化程式設計工作。例如在Java語言中，可將正常執行的指令寫在一個try區段中，而當這些指令執行出現錯誤或例外時，希望用來處理錯誤或例外的指令則寫在try區段後面的各個catch子句中，如下列程式碼所示：

```
try {
    OpenSocket ();
    OpenStream ();
    ReadStream ();
    //Process stream data
} catch (ReadStreamError) {
    //Handle read stream error
} catch (OpenStreamError) {
    //Handle open stream error
} catch (OpenSocketError) {
    //Handle open socket error
}
```

當try區段中的程式指令正常執行時，各catch子句皆無作用；但若有某個指令在執行時出現錯誤 (例外) 情況，系統將根據例外的型態自動從try區段後面找出一個對應的catch子句作為處理程式，以處理所出現的錯誤 (例外)。

二、(一)何謂資料抽象化 (data abstraction) ? (10分)

(二)資料抽象化在程式語言之功能為何? 舉例說明之。(10分)

【擬答】

(一)資料抽象化 (data abstraction) 是指可將要處理的資料及作用在該資料上的各個運算包藏在一個語法單元中的特性稱之。

(二)資料抽象化在程式語言中可運用其包藏 (encapsulation) 和資訊隱藏 (information hiding) 的功能，促進以資料為中心之模組化，並可避免所處理的資料遭到隨意之破壞，如此可大大簡化複雜或龐大程式之設計工作。例如在C++語言中，可將組成堆疊 (stack) 的陣列array和頂端註標top等資料連同處理堆疊的push和pop等運算包藏在一個類別 (class) 語法單元中，程式碼如下：

```
#include <iostream.h>
class Stack {
private:
    int array[100];
    int top;
public:
    Stack () { top = -1; }
    void push(int element)
    {
        if ( top < 99 ) {
```

```

        top = top + 1;
        array[top] = element;
    }
    else
        cout << "Stack full !\n";
}
int pop(void)
{
    if ( top >= 0 )
        return array[top--];
    else {
        cout << "Stack empty !\n";
        return -999;
    }
}
};
void main(void)
{
    Stack    aStack;                //宣告堆疊物件
    aStack.push(10);                //存入資料10
    aStack.push(20);                //存入資料20
    aStack.push(30);                //存入資料30
    cout << aStack.pop() << endl;    //刪除並輸出資料30
    cout << aStack.pop() << endl;    //刪除並輸出資料20
    cout << aStack.pop() << endl;    //刪除並輸出資料10
}

```

在此程式中，整個Stack類別可視為一個完整的程式單元，而陣列array與top註標因定義在類別的private區域，它們將被隱藏且受到保護，外界（如主程式main中）將無法直接存取到它們，只能透過此類別所提供的push和pop運算來處理堆疊資料。

三、請以任何一程式語言寫一程式來處理河內塔（Tower of Hanoi）問題，並請使用遞迴副程式（recursive call，即副程式呼叫本身）。河內塔（Tower of Hanoi）問題是一個著名的遞迴問題。假設有n個套環（disc），大小都不相同，編號為1, 2, ..., n（依由小而大的次序）。最初此n個套環都在A柱，今欲將此n個套環搬到C柱，規定一次只能搬動一個套環，且小套環永遠在大套環之上，另有B柱當輔助。（20分）

【擬答】

以C++撰寫程式如下

```

#include    <iostream.h>

void Hanoi(int n, char Source, char Dest, char Aux)
{
    if ( n == 1 )
        cout << "Move disk 1 from " << Source << " to " << Dest << endl;
    else {
        Hanoi(n-1, Source, Aux, Dest);
        cout << "Move disk " << n << " from " << Source << " to " << Dest << endl;
        Hanoi(n-1, Aux, Dest, Source);
    }
}

void main(void)
{
    int    n;
    cout << "Enter an integer : ";
}

```

```
cin >> n;
Hanoi(n, 'A', 'C', 'B');
}
```

執行結果：

```
Enter an integer : 3
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
```

四、試算出下列虛擬指令之時間複雜度

(一)for i = 0 to n do

begin

j = i;

while j > 0 do j = j - 1;

end (10分)

(二)for (x = 1; x < n; x++)

{...} (10分)

【擬答】

(一) $O(n^2)$

(二) $O(n)$

五、考慮下述文法：

$\langle S \rangle \rightarrow \langle A \rangle a \langle B \rangle b$

$\langle A \rangle \rightarrow \langle A \rangle b \mid b$

$\langle B \rangle \rightarrow a \langle B \rangle \mid a$

則下述那些句子可由上述文法產生，並說明之。

(一)bbaabb

(二)bbaaaa

(三)bbbab

(四)baab

【擬答】

此文法只能產生語句baab，其推導過程如下：

```
<S>    ⇒ <A>a<B>b
        ⇒ ba<B>b
        ⇒ baab
```

由於從 $\langle S \rangle$ 所導出來的語句都是由 $\langle A \rangle$ 開始，並以b結束，且 $\langle A \rangle$ 只能產生一至多個b，而 $\langle B \rangle$ 只能產生一至多個a，因此從 $\langle S \rangle$ 所導出來的語句中最後只有一個b，且b前面至少兩個a，故bbaabb、bbaaaa及bbbab由此文法產生出來。