

105 年公務人員特種考試關務人員考試試題

考試別：關務人員考試

等別：四等考試

類科：資訊處理

科目：程式語言概要

- 一、(一)請說明物件導向式程式語言(object-oriented programming language)三大特色。(15分)
(二)許多物件導向式程式語言允許程式員定義物件的建構函數(constructor)與解構函數(destructor)，請說明建構函數與解構函數的用途。(10分)

【擬答】：

(一)物件導向式程式語言(object-oriented programming language)三大特色

1. 物件：提供資料和處理資料程序的封裝。有以下特性

(1) 狀態 (State)：物件所有「屬性」(Attributes)目前的狀態值，屬性是儲存物件的狀態，可以簡單到只是一個布林值變數，也可能是另一個物件。例如：車子的車型、排氣量、色彩和自排或手排等屬性，以程式來說，也就是資料部分的變數。

(2) 行為 (Behavior)：行為是物件可見部分提供的服務，也就是塑模所抽象化的操作，可以做什麼事？程式語言是使用程序或函數來實作行為，稱為方法，例如：車子可以發動、停車、加速和換擋等。

(3) 識別字 (Identity)：每一個物件都擁有獨一無二的識別字來識別它是不同的物件。例如：使用物件參考 (Reference) 作為物件的識別字，即物件實際儲存的記憶體位址。

2. 訊息：在物件之間的溝通方式，可以建立互動和支援多形。物件是使用訊息來模擬彼此之間的互動，換句話說，訊息是物件之間的溝通橋樑，可以啟動另一個物件來執行指定的行為。

3. 類別：物件的分類，可以實作類別架構的繼承。類別是具有相同資料結構及行為的物件所組成的集合。一個類別是一種定義(Definition)、樣板(Template)或模子(Mold)，它是許多物件共同特徵的一種描述。類別是物件經分類與抽象化後所得的結果。將物件抽象化，就是剔除物件間的差異，而只考慮其相同的性質，然後組成一個群體，稱為類別。

(二)建構函數與解構函數的用途

1. 建構函數 (constructor)

(1) 是類別的函數成員。

(2) 與類別的名稱相同。

(3) 無傳回值，但不需加 void。

(4) 當建立一個物件時會自動被呼叫。

建構函數的使用

```
#include<iostream . h>
#include<string . h>
class student
{
public :
char name [10] ;
char sex ;
student (char * , int) ; // constructor
void get ( ) ;
private :
int age ;
char tel [7] ;
```

```
};  
5student :: student (char * tel , int age)  
{  
    if ( age >= 18 ) student :: age = age ;  
    else          student :: age = 0 ; // age < 18  
    strcpy (student :: tel , tel);  
}  
void student :: get ( )  
{  
    cout << "Name : " << name << endl ;  
    cout << "Sex : " << sex << endl ;  
    cout << "Age : " << age <<endl ;  
    cout << "Tel : " << tel << endl ;  
}  
void main ( )  
{  
    student john ("1234567" , 20) ;  
    strcpy (john . name , "John Lin") ;  
    john . sex = 'M' ;  
    john . get ( ) ;  
}
```

執行結果

```
Name : John Lin  
Sex : M  
Age : 20  
Tel :1234567
```

2. 解構函數 (destructor)

使用解構函數來釋放這些記憶體。解構函數的名稱與類別名稱相同，但其前要加上~符號。

解構函數不須使用參數。

解構函數的使用

```
#include<iostream . h>  
#include<string . h>  
class student  
{  
    public :  
        char name [10] ;  
        char sex ;  
        student (char * , int ) ; // constructor  
        ~student ( ) ; // destructor  
        void get ( ) ;  
    private :  
        int age ;  
        char tel [7] ;  
};  
student :: student (char * tel , int age)  
{  
    if ( age >= 18 ) student :: age = age ;
```

公職王歷屆試題 (105 關務特考)

```
else student :: age = 0 ; // age < 18
strcpy (student :: tel , tel ) ;
}
void student :: ~student ( )
{
    cout << "Destroying the object for" ;
    cout << student :: name << endl ;
}
void student :: get ( )
{
    cout << "Name: " << name << endl ;
    cout << "Sex : " << sex << endl ;
    cout << "Age : " << age << endl ;
    cout << "Tel : " << tel << endl ;
}
void main ( )
{
    student john ( "1234567" , 20) ;
    strcpy ( john . name , "John Lin" ) ;
    john . sex = 'M' ;
    john . get ( ) ;
}
```

執行結果

```
Name: John Lin
Sex: M
Age : 20
Tel:1234567
Destroying the object for John Lin
```

二、根據下列的兩個程式：

程式(A)prolog

```
fooA(1,1)
fooA(N,X):-M is N-1, fooA(M,Y), X is 3*Y+1.
```

程式(B)python

```
def fooB(n):
    a, b=1, 3; k=1
    while k<=n:
        a, b=b, a+b; k=k+1
    print( 'fooB(%d)= ' %n, b)
    return
```

(一)請說明程式(A)與程式(B)分別是用下列那一種語言所撰寫。(6分)

(二)請說明程式(A)與程式(B)所使用的那兩種語言各自偏向那一種程式語言範疇(paradigm)(例如 procedural programming language、dataflow programming language、...等)。(8分)

(三)針對程式(A)，若執行 fooA(3, X)，X 的值會是 13。當執行 fooA(5, X)，X 的值為何？(5分)

公職王歷屆試題 (105 關務特考)

(四)針對程式(B)，若執行 `fooB(1)`，會輸出：

`fooB(1)=4`

那麼執行 `fooB(7)`，輸出為何？(6分)

【擬答】：

(一)C++、Java、ML、Prolog、Python、Ruby

程式 A 是 prolog 偏人工智慧；程式 B 是 python 是一種物件導向、直譯式的電腦程式語

10

1

三、根據變數的生命週期(lifetime)，請針對下列 C 程式中的 5 個物件包括變數 `w, x, y, z` 與指標 `z` 所指結構物件，分別說明它們是使用何種記憶配置(allocation)方式。(25分)

```
#include<stdlib.h>
typedef struct {int u,v;} point;
int w;
Point*foo(void)
{
    int x;
    static int y;
    point *z;
    z=(point*)malloc(sizeof(point));
    /* .... */
    Return z;
}
```

【擬答】：

(一)區域變數(local variable)的特性

1. 函數內的變數含輸入參數預設為區域變數。
2. 區域變數的生命週期只存在函數執行期間，當該函數執行完畢，函數內的區域變數即會從 workspace 或記憶體中消失。

函數外其他程式不能讀取函數內的任何區域變數，除非函數內的區域變數設定為全域變數。

(二)全域變數(global variable) 的特性

1. 存在 workspace 的變數均為全域變數。
2. 所有的程式都有權讀取全域變數，區域變數的讀取權僅限於所屬函數。
3. 全域變數的生命週期從該變數產生開始即會存在於 workspace，直到離開 matlab 主程式或執行 clear 指令，全域變數才會從記憶中消失。

`w` 是靜態配置；`x` 是動態配置；`y` 是靜態配置；`z` 是靜態配置；`*z` 是外顯堆積動態變數

四、考慮下列的語法。

$\text{expr} \rightarrow \text{term} \mid \text{expr} * \text{term}$

$\text{term} \rightarrow \text{id} \mid \text{term} + \text{id}$

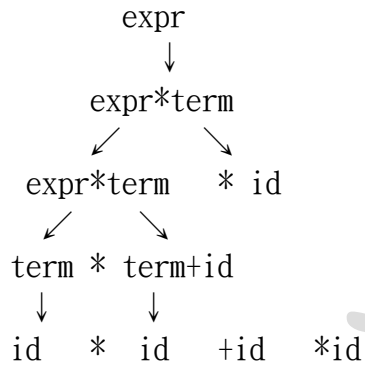
$\text{id} \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

(一)請根據上述語法，畫出 $1*2+3*4$ 的剖析樹(parse tree)。(15分)

(二)請改寫上述的語法，讓 * 運算子的優先等級(operator precedence)高於 + 運算子。並以 $1+2*3$ 的剖析樹為例，證明改寫後的語法， $1+2*3$ 解釋為 $1+(2*3)$ ，而不是 $(1+2)*3$ 。(10分)

【擬答】：

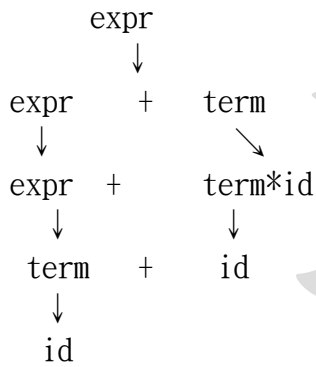
(一)



(二)

$expr \rightarrow term \mid expr+term$
 $term \rightarrow id \mid term*id$
 $id \rightarrow 0|1|2|3|4|5|6|7|8|9$

所以 $1+2*3$



沒有產生模擬兩可的語法