

《程式語言》

試題評析	<p>今年試題整體而言難度不高。</p> <p>第一題和今年高考題目類似，屬於上課講義 4-3,4-4 章節。</p> <p>第二題是基本題型，和去年高考題目類似，在上課講義第一章有詳細講解。</p> <p>第三題考名詞解釋，對熟悉網路腳本語言同學較有利。</p> <p>第四題考函數型程式語言的概念，對有練習講義第二章、有補充講義第三回的同学應屬不難。</p> <p>第五題考基本參數傳遞方法，在講義 8-2 有詳細講解。</p> <p>第六題為基本遞迴程式撰寫，為補充講義第一回的範疇。</p> <p>預估一般分數落在 65~75 分，有網路腳本語言實務撰寫經驗的同學可以拿到 80 分以上。</p>
------	--

- 一、(一)何謂類別化的程式語言 (Typed Programming Language) ? (5 分)
- (二)在類別化的程式語言中，何謂動態的類別化 (Dynamic typing) ? (5 分)
- (三)何謂強烈的類別化 (Strong typing) ? (5 分)

【擬答】

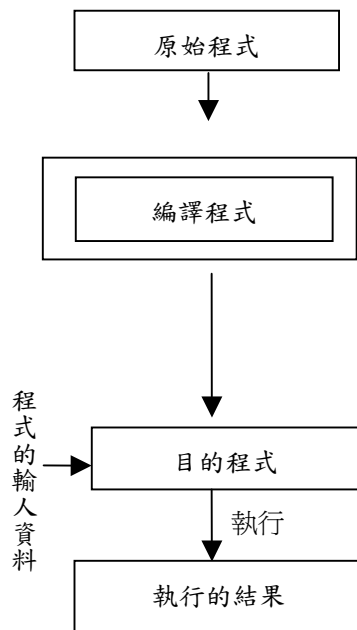
- (一)類別化程式語言(Typed programming language)指的是程式語言本身具有自己定義的類別系統(type system)，包括：
1. 語言本身定義的資料型態(例如：整數,浮點數,字元等)。
 2. 使用者自訂資料型態的語法與限制(例如：C++利用 struct/class 定義資料型態)。
 3. 檢查資料型態是否相容的規則(例如：字串資料能不能和整數相加)。
 4. 特定資料型態可以做的運算(例如：整數允許做加減乘除等數學運算)。
 5. 決定資料型態的方式(例如：動態或靜態型態繫結)。
- (二)動態類別化(Dynamic typing)是指變數的資料型態到執行時期(動態)依照其實際值決定。例如：Lisp 程式語言中變數不需宣告指定其型態，變數型態是執行時期(動態)依照指定給變數的值來決定。
- Ex：setq(a 3) 變數 a 的型態在執行時期決定為整數(內容為整數 3)。
- (三)強烈類別化(Strong typing)是指：
1. 每一個變數都靜態地繫結到單一資料型態。
 2. 變數資料型態繫結後在程式執行過程不能更改其資料型態。
 3. 某一運算中所用到的兩個變數可以靜態地檢查出它們的相容性。
 4. 當允許一個變數儲存不同資料型態之值時，其值的型態可以靜態地或動態地檢查出來。
 5. 不允許編譯器對變數做隱性形態轉換，只能顯示形態轉換。

- 二、(一)在電腦中，程式碼 (source code) 的執行 (execution) 有那兩種方法? (4 分)
- (二)請說明它們各別的優缺點。(6 分)

【擬答】

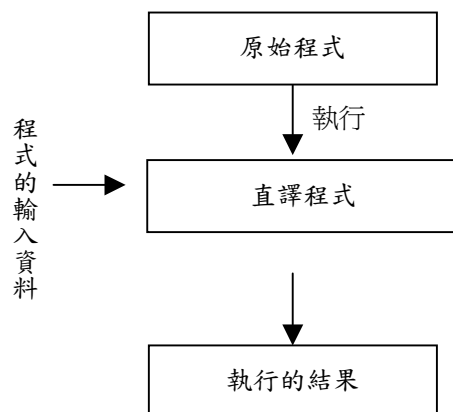
- (一)1.編譯(compilation)
- 將高階語言(例如：C++，Java，Pascal 等各種程式語言)所寫之程式(source program)的全部敘述轉換成能直接被機器接受之等效目的程式(object program)。





2. 直譯(interpretation)

按照高階語言(例如：Java script)所寫的程式執行時敘述的邏輯順序，逐一指令轉為機器語言指令並且執行之。



(二) 1. 編譯程式的優缺點

優點：

(1) 目的程式執行階段效能較高：

每次執行時只要把已編譯完成之目的程式載入記憶體直接執行即可，不必再進行原始程式的解碼。

(2) 節省解碼的時間：

編譯程式按照原始程式實體上的輸入順序進行解碼，每個敘述都只解碼一次，執行時則直接執行產生之目的程式，因此即使某些敘述重覆執行許多次也不須再解碼，所以節省了許多的解碼時間。



(3)程式可以進行最佳化：

編譯器依照原始程式特性或是硬體特性進行最佳化。

缺點：

(1)執行時需要比較大的記憶體空間：

原始程式中單一個敘述經過編譯程式解碼轉換後，可能產生成千上百個機器語言指令，編譯程式將原始程式內所有的指令都給予轉換，產生目的程式存放到輔助儲存體(例如：磁碟)；欲執行該程式時，由載入程式將目的程式載入記憶體後執行之，因此執行時需要比較大的記憶體空間。

(2)儲存時需要比較大的輔助儲存體空間：

將原始程式及目的程式一併儲存，因此需要比較大的輔助儲存體空間。

(3)執行階段發生錯誤之處理比較費時及麻煩：

需將原始程式修改，然後重新編譯產生目的程式，再重新執行。

(4)互動性較差：

編譯程式是把程式完全翻譯成機器碼後執行。

(5)產生的執行檔不能跨平台執行：

編譯程式會產生特定機器平台的機器碼，執行檔不具可攜性。

2.直譯程式的優缺點

優點：

(1)執行時所需要的記憶體空間比較小：

直譯程式按照原始程式執行時敘述的邏輯順序逐一地解碼轉換並且執行，而不是等到原始程式中所有指令都轉換後才執行，因此需要的記憶體空間比較小。

(2)儲存時需要的輔助儲存體空間比較小：

只要將原始程式儲存即可。

(3)執行階段發生錯誤之處理比較簡單而節省時間：

可直接修改敘述或資料後繼續轉換及執行。

(4)互動性較高：

直譯程式是邊翻譯邊執行

(5)程式碼具可以在不同機器平台上執行：

只要特定機器平台上有此語言的直譯器

缺點：

(1)執行階段效能低：

每執行次直譯程式都將原始程式直接輸入，然後依照執行時敘述的邏輯順序逐一地解碼轉換並且執行，因而花費許多時間在敘述的解碼上，因此執行所需要的時間比較長。

(2)解碼所需要的時間比較長：

直譯程式將原始程式直接輸入，然後依照執行時敘述的邏輯順序逐一地解碼轉換並且執行，若某些敘述執行許多次，則每次都需先進行解碼轉換才執行，因此需要比較長的解碼時間。

(3)無法對程式做最佳化處理。

三、(一)在網頁的設計中，何謂一個動態的網頁文件？(5分)

(二)舉例說明動態網頁文件如何製作。(10分)

【擬答】

(一)動態網頁文件利用資料庫來儲存網頁資料，並且透過網頁伺服器端的即時處理網頁腳本程式語言的需求，轉譯出用戶端要求的網頁，並且傳送至用戶端瀏覽器呈現，包含以下主要特性：

1.使用資料庫來儲存網頁相關資料。

2.利用網頁腳本程式語言(script language)，例如 Java script，ASP，PHP 等去存取網頁伺服器端資料，伺服器將此需求轉換成對應的 HTML 本文檔送回去給要求的瀏覽器。

3.動態網頁"不是"指有"動態視覺效果"的網頁，例如：圖片會變換的網頁不算是動態網頁。



高點律師司法官班 <http://www.license.com.tw/lawyer/>
 北市開封街一段 2 號 8 樓 • 02-23115586 (代表號)

(二)製作動態網頁文件的流程

- 1.安裝網頁伺服器，例如：Apache。
- 2.安裝資料庫軟體，例如：MySQL。
- 3.安裝網頁腳本程式語言環境，例如：PHP。
- 4.撰寫 HTML 本文檔。
- 5.在 HTML 本文檔中加入網頁腳本語言，例如：PHP 來連結網頁伺服器/資料庫端。
- 6.將寫好的網頁放到伺服器上特定目錄下讓用戶端存取。

四、假設 f 是一個函數， $inarray$ 是一個一維的陣列 (array)，定義 $Map(f, inarray)$ 的功用是將 f 這個函數應用到 $inarray$ 陣列的每個元素，然後輸出這個結果陣列。例如 $Map(\text{function}(x)\{\text{return } x+1\}, [1, 2, 3, 4, 5])$ 會得到 $[2, 3, 4, 5, 6]$ ， $Map(\text{function}(x)\{\text{return } 2x\}, [1, 2, 3, 4, 5])$ 會得到 $[2, 4, 6, 8, 10]$ 。假設 g 是一個有兩個輸入參數 (argument) 的函數，其結果為一個元素，例如 $g(y, z)\{\text{return } (y+z)\}$ 即是得到兩數的和。定義 $Reduce(g, inarray)$ 為將 g 應用到陣列 $inarray$ 的元素，由左至右，直到最後成為一個元素為止，例如利用上述的 g ， $Reduce(g, [1, 2, 3, 4, 5])$ 的結果為 15，若 $g=g(y, z)\{\text{return } y*x\}$ ，則 $Reduce(g, [1, 2, 3, 4, 5])$ 的結果為 120。

(一)請用虛擬指令 (pseudo code) 的方式，寫出 Map 函數的程式。(10 分)

(二)請用虛擬指令 (pseudo code) 的方式，寫出 $Reduce$ 函數的程式。(10 分)

(三)請用 Map 和 $Reduce$ 兩個函數，說明如何計算 $1^2+2^2+3^2+4^2+5^2$ 。(10 分)

【擬答】

(一)假設陣列元素註標由 0 開始

```
ElementType array[] Map(function f, ElementType inArray[])
{
    ElementType outArray[elementNumOf(inArray)];
    for(int i = 0; i < elementNumOf(outArray); ++i) {
        outArray[i] = f(inArray[i]);
    }
    return outArray;
}
```

(二)假設陣列元素註標由 0 開始

```
ElementType Reduce(function f, ElementType inArray[])
{
    if (elementNumOf(inArray) <= 1) {
        return f(inArray[0], ElementType(0));
    } else {
        ElementType result = f(inArray[0], inArray[1]);
        for (int i = 2; i < elementNumOf(inArray); ++i) {
            result = f(result, inArray[i]);
        }
        return result;
    }
}
```



```
(三)int square(x) = {
    return x * x;
}
int add(int num1, int num2)
{
    return num1 + num2;
}
void main()
{
    //計算 1~5 的平方和
    int inArray[] = {1, 2, 3, 4, 5};
    print Reduce(add, Map(square, inArray));
}
```

五、在下列的虛擬程式碼中：

```
Function f(x, y: integer)
{x=2*x; y=x+y; return (x+y)}
```

假設 a=1, b=2, 請問

(一)若參數的傳遞是靠 call-by-value, 請問 f(a, b) 的值是多少? 執行完 f(a, b) 後, a=? , b=? (10 分)

(二)若參數的傳遞是靠 call-by-reference, 請問 f(a, b) 的值是多少? 執行完 f(a, b) 後, a=? , b=? (10 分)

【擬答】

(一) Call-by-value : $f(a,b) = x + y = 2 + 4 = 6$
a = 1, b = 2

(二) Call-by-reference: $f(a,b) = x + y = 2 + 4 = 6$
a = 2, b = 4

六、將下列的遞迴 (recursive) 程式改成非遞迴的版本: (10 分)

```
long f(n: integer)
{if(n<=1) return (1);
else return n*f(n-1); }
```

【擬答】

```
long f(n: integer)
{
    int fact = 1;
    for (int i = 1; i <= n; ++i) {
        fact *= i;
    }
    return fact;
}
```

