

# 《程式語言》

一、請試述下列名詞之意涵：（每小題3分，共9分）

- (一)Dead lock
- (二)NP-Complete Problem
- (三)Cookie

試題評析	本題為名詞解釋，前面二題為作業系統與資料結構相關名詞，以往出現在系統程式或資料結構考科中，在程式語言中較少出現。若考生同時準備資料結構，應不難應答，唯未來仍須注意相關題目的趨勢。
考點命中	1.《高點·高上程式語言講義》第三回，金乃傑編撰，頁124及課堂補充。 2.《高點·高上資訊管理講義》第一回，金乃傑編撰，頁53-54及課堂補充。

答：

- (一)Dead lock：死結，為兩個以上的運算單元，雙方都在等待對方停止執行，以取得系統資源，但是沒有一方提前退出時所造成鎖死的現象。如果系統中只有一個運算單元或每個運算單元需要的系統資源不同就不會產生死結，根據研究，形成死結的四個條件為：禁止搶奪（no preemption）、持有且等待（hold and wait）、互斥（mutual exclusion）與循環等待（circular waiting），若能破壞其中任一因素，即可解除死結。
- (二)NP-Complete Problem：NP-Complete是計算複雜度理論中的一種等級，而NP-Complete Problem，是NP中最難的決定性問題，因此NP-Complete Problem應該是最不可能被化簡為多項式時間可決定的問題集合。例如：給定一個有限的整數集合，找出所有此集合中和為0的子集合，且子集合非空。這個問題雖然容易驗證，但目前只能一個個將子集取出來測試，沒有夠快的方法可以在合理的時間內（意即多項式時間）找到答案，因此時間複雜度是 $O(2^n)$ ， $n$ 是此集合的元素數量。
- (三)Cookie：為小型文字檔案，是網站為了辨別用戶身分而儲存在用戶端（Client Side）上存放的資料，定義於RFC2109，為網景公司的前員工Lou Montulli在1993年3月所發明。因為HTTP協定的伺服器不知道使用者上次的操作，限制互動式Web應用程式的開發。例如：在網路購物時，使用者買了一些商品，但在結帳時由於HTTP協定無法記憶狀態，伺服器並不知道用戶到底買了什麼。伺服器可以設定或讀取Cookies中包含資訊，藉此維護用戶跟伺服器對談中的狀態。

二、AlphaGo擊敗人類高段的職業圍棋棋士，主要是利用深度學習技巧，也讓深度學習已成為臺灣未來科技重要產業，請回答下列問題：（每小題4分，共20分）

- (一)何謂神經網路？
- (二)何謂深度學習(deep learning)？
- (三)神經網路因為架構問題無法做到很多層，為何Deep learning可以做到很多層？
- (四)何謂CNN(convolutional neural network)？
- (五)在深度學習裡，除了C以外，另一種常用的語言為Python，請試述Python跟C有何差別？

試題評析	本題屬於時事題，內容以年初的AlphaGo下棋及「深度學習」趨勢，亦屬於新增考題。由於以往相關知識領域如：機器學習、資料探勘、類神經網路等主要出現在資訊管理考科中，因此雖程式語言中無正式介紹，但若準備高考，則必會同時準備到資訊管理，因此對同學而言應不生疏。但值得注意的是，本題亦是三四等程式語言考試中正式出現Python語言的考題，雖然配分不高，但由於Python與LISP同屬函數型程式語言，在函數型程式語言考題缺乏時亦可能加入題型中，因此與先前出現C#後不了了之的情況不同，在未來準備國家考試時亦須對於Python語法需要有初步的認識。關於深度學習相關的運作介紹亦可參考此網站： <a href="http://pics.ee/1KTU">http://pics.ee/1KTU</a>
考點命中	《高點·高上資訊管理講義》第一回，金乃傑編撰，頁53-54及課堂補充。

答：

- (一)神經網路：又稱類神經網路(Artificial Neural Network, ANN)，以程式模擬大腦的學習過程，程式架構分為輸入、隱藏與輸出三層，透過連線中權重的改變達到機器學習，建立資料探勘的分類機制。類神經網路的執行時可分為訓練與測試2階段，可透過過去案例來學習、修正其解決問題的能力。
- (二)深度學習：是機器學習(Machine Learning)中近年來備受重視的，深度學習根源於類神經網路模型。深度學習可看成是一種資訊的表達方式，而「多層次神經網路」則是最適合實現這種表達方式的技術。多層次神經網路在最底層把資料分散成單元，我們可以把每個單元想成是一個神經元，這些神經元經過組合的結果，成為第二層的輸入，如此一層一層訓練出來的機器學習模組，辨識能力將越來越強。目前深度學習在語音辨識和影像辨識上表現最好。
- (三)傳統類神經網路僅三層：輸入層、隱藏層及輸出層。因為以往每個神經元的初始值採隨機設定，搭配倒傳遞(back propagation)使用已標籤的資料修正各神經元中的權重。此設計最大的問題是若層次太多，修正時間長，且因為各層次中殘差在每一層被稀釋，傳到最上層時會變得太小，出現梯度擴散(gradient diffusion)，因此難以藉由比對標記(label)好的資料來修正權重。但深度學習在初始化時並非使用亂數，而是使用非監督學習的方式，並且以分層訓練，一層計算完再以輸出作為下一層的輸入，在每一層中讓系統自己找到特徵(feature learning)。最後再反過來由上往下以標籤過的資料加以訓練，微調權重，因為是分層訓練(layer-wise)，因此可以避免層次過多梯度擴散的問題。
- (四)CNN(Convolutional Neural Network, CNN)：卷積神經網路，是一種深度學習的技術，對於大型圖像處理有出色表現。它由一或多個卷積層和頂端的全連通層所組成，也包括關聯權重和池化層(pooling layer)，此結構使卷積神經網路能夠處理二維的數據，由於需要估計的參數更少，使之成為一種頗具潛力的深度學習結構。
- (五)將Python與C語言差異比較如下：

	Python	C語言
特性適用	以Web應用程式為主，對於各種網路協定的支援完善，經常用於編寫伺服器軟體、網路蠕蟲。	高效能、直接與底層硬體溝通，常用於設計作業系統、要求高效能、關鍵系統或與硬體溝通。
執行	直譯語言，可在直譯器中直接輸入指令執行。	編譯語言，需先將程式原始碼編譯後才能執行
變數	不需要宣告即可使用，提供內隱堆積動態變數，根據變數實際內容型態可變化。	使用變數前須先宣告，提供靜態變數、堆疊動態變數與外顯堆積動態變數。
函數	使用函數前須先宣告，但提供匿名函數，函數不須指定回傳型態。	使用函數前須先宣告，須設定回傳型態，若回傳型態不為void，則須設置回傳值。
迴圈	支援for、while兩種。	支援for、while與do-while三種。
例外處理	提供，使用try語法，需與except、finally、else配合使用。	不支援。
垃圾回收	支援垃圾回收功能，能夠自動管理記憶體使用。	不支援，記憶體需手動釋放。
物件導向	支援，以class關鍵字宣告。	不支援。
其他	具有強大的函式庫，第三方函式庫資源也越來越多，而直譯器幾乎可在任何作業系統中執行。	現成函式庫較少，在不同平台上需要重新編譯才能使用，跨平台特性較差。

三、利用二分法搜尋(binary search)，從100000筆資料中，搜尋某一特定資料，最多會比對幾次？(5分)

**試題評析** 本題屬於資料結構中搜尋時間複雜度相關考題，不予評論。

**答：**

二分搜尋法最差情況為 $O(\log_2 n)$ ，故100,000筆資料最多需比對17次。

$\log_2 100000 = 16.6096\dots$ ，還需對找到的元素再比對一次，故需17次， $2^{17} = 131,072$

四、試就Object Oriented language與Functional language，回答下列問題：

(一)比較兩種語言之不同與優缺點。(5分)

(二)續題(一)，請對下列五種語言做歸類：C、Java、C++、Lisp、Prolog，並以下表作答於試卷上(如果不屬於這兩類語言，請在None打勾)。(10分)

語言	Object Oriented language	Functional language	None
C			
Java			
C++			
Lisp			
Prolog			

<b>試題評析</b>	本題為不同程式語言比較題，第一小題以物件導向與函數型程式語言的優缺點作比較，而第二小題要求把程式語言以物件導向、函數型歸類。值得注意的是這是近年來對函數型程式語言較重視的一張考卷，可能是未來考試趨勢，同學在準備時亦須要對不同語言的類型、特性加以了解，才能獲得高分。
<b>考點命中</b>	1.《高點·高上程式語言講義》第五回，金乃傑編撰，頁1。 2.《高點·高上程式語言講義》第六回，金乃傑編撰，頁65-70。

**答：**

將Object Oriented language與Functional language之優缺點比較如下：

	Object Oriented language	Functional language
<b>優點</b>	1.封裝(Encapsulation)：將資料及操作包在類別中，有助於資訊隱藏(Information hiding)及撰寫大型系統。 2.繼承(Inheritance)：父類別的資料和操作可供子類別使用，增加再用性(reusability)。 3.多型(Polymorphism)：執行時期根據物件實際內容動態決定使用的方法，使相同程式碼會有不同的執行狀態。	1.易於學習、撰寫：由函數構成，語法、語意均較容易。 2.易於開發：直譯器執行可以馬上除錯、知道結果。 3.參考透明性(Referential Transparency)：無邊際效應，給予相同的參數，就會有相同的結果。
<b>缺點</b>	1.程式結構較複雜，需宣告類別再產生物件才能執行。 2.需考慮記憶體洩漏(Memory Leakage)問題，建立物件對物件的記憶體管理成為負擔。	通常為直譯語言，執行效率較差，不適用於大型系統。

依照題意完成表格如下：

語言	Object Oriented language	Functional language	None
C			V
Java	V		
C++	V		
Lisp		V	
Prolog			V

五、C語言宣告一陣列如int \*x[5]={"BENZ", "AUDI", "LEXUS", "APPLE", "BMW"}；試說明下列指標運算的結果為何？(每小題3分，共9分)

(一)\*(x+1)

(二)\*(x+1)+1

(三)\*\*(x+1)

試題評析	本題為陣列與指標考題，相似於101普考、100高考與98高考。若同學對於陣列概念有掌握，應可獲得高分。
考點命中	《高點·高上程式語言講義》第三回，金乃傑編撰，頁11-16。

**答：**

(一)AUDI

(二)UDI

(三)A

說明：

題目宣告有誤，應改為 `char *x[5]`，不然無法執行。

可使用以下程式測試：

`printf("%s\n", *(x+1));``printf("%s\n", *(x + 1)+1);``printf("%c\n", ***(x + 1));`

六、在程式語言中變數可歸類成下列幾種:static variable、stack-dynamic variable、explicit heap-dynamic variable、implicit heap-dynamic variable。

(一)請比較上述變數的差異。(12分)

(二)在C++語言裡，如果有變數定義如下`int *data; data=new int; delete datd;`請問“data”屬於哪一種變數？試說明之。(3分)

試題評析	本題第一小題比較程式語言中儲存體繫結（變數在記憶體中的位置）的差異，為104鐵路與102檢事官的相似題；第二小題測驗同學對於程式語法的掌握，是否能解釋屬於何種儲存體繫結。若考生對歷屆考題充分練習，此題應可獲得高分。
考點命中	《高點·高上程式語言講義》第三回，金乃傑編撰，頁69-70, 73-74。

**答：**

(一)依照題意將static variable、stack-dynamic variable、explicit heap-dynamic variable及implicit heap-dynamic variable之差異比較如下表：

種類	static variable	stack-dynamic variable	explicit heap-dynamic variable	implicit heap-dynamic variable
繫結時間	載入階段，程式執行前已繫結到記憶體。	執行階段，進入副程式活動紀錄被建立時。	執行階段，執行動態配置記憶體敘述時建立。	執行階段，執行變數指定敘述時建立。
結束時間	程式結束	副程式歸還記憶體。	程式結束或手動釋放。	垃圾回收機制啟動或程式結束。
優點	1.執行效率佳。 2.支援歷史相關特性。 3.提供函數、類別間溝通。	1.支援遞迴程式。 2.節省記憶體。	支援動態結構。	撰寫彈性極高。
缺點	1.記憶體運用缺乏彈性。 2.不支援遞迴程式。	1.執行時必須配置、釋除記憶體。 2.不支援歷史相關特性。	1.容易產生人為錯誤。 2.執行效率較差。	1.編譯器偵錯成本高。 2.執行效率較差。
實例	全域變數。 靜態變數。	區域變數。	new、delete malloc()、free()	\$a = 5; \$a = "no".5;

(二)stack-dynamic variable，依照題目data之宣告，採用int \*，故可知data為一可存放記憶體位址的變數，而此變

數指向一塊手動建立(new)且可釋放(delete)的記憶體空間，該空間為explicit heap-dynamic。值得一提的是，若此語言為Java，可透過參考直接指向該種記憶體空間，而非透過指標間接定址，則data才有可能屬於explicit heap-dynamic variable。

七、請回答下列問題：(每小題5分，共10分)

- (一)給定一個整數陣列S [ n ]，請寫出一個副程式 int SelectionK(int\*S, int n)，此函數可以回傳(return)第K大的數值。
- (二)給定一個陣列S [ n ]，請寫出一個演算法，此演算法可以用平均時間複雜度為O(n)的效率，回傳(return)第K大的數值。

試題評析	本題題目有瑕疵，因若需回傳陣列中最大值，則必須要透過排序，不然時間複雜度必為 $O(n^2)$ ，故第一小題撰寫程式時可考慮先透過排序演算法排序後，直接透過陣列參考指向第K大數值，而此題要設計函數便失去意義。同理，第二小題之演算法亦不可能實現，有失國家考試水準。
考點命中	《高點·高上程式語言講義》第一回，金乃傑編撰，頁103-105。

答：

(一)假設函數增加參數k變為SelectionK(S, n, k)，以指定需要找尋的k，以C++撰寫如下：

```

6 //合併排序法，時間複雜度O(nlogn)，請參考資料結構
7 void mergeSort(int *data, int n){
8     int i, leftMin, leftMax, rightMin, rightMax, next;
9     int *tmp = new int[n];
10
11     for (int i = 1; i < n; i *= 2){
12         for (leftMin = 0; leftMin < n - i; leftMin = rightMax){
13             rightMin = leftMax = leftMin + i;
14             rightMax = leftMax + i;
15             if (rightMax > n) rightMax = n;
16             next = 0;
17             while (leftMin < leftMax && rightMin < rightMax)
18                 tmp[next++] = (data[leftMin] > data[rightMin])?
19                 data[rightMin++]: data[leftMin++];
20             while (leftMin < leftMax) data[--rightMin] = data[--leftMax];
21             while (next > 0) data[--rightMin] = tmp[--next];
22         }
23     }
24 };
25
26 int SelectionK(int *S, int n, int k){
27     mergeSort(S, n); //先排序
28     return S[n - k]; //直接指到第k大
29 }
30
31 int main(){
32     int S[] = { 7, 17, 2, 11, 13, 3, 5, 19 };
33     int n = sizeof(S) / sizeof(int); //計算陣列個數
34     int k = 5; //假設搜尋第5大
35
36     int nk = SelectionK(S, n, k);
37     printf("%dth large num is: %d", k, nk);
38 }

```

輸出如下：

5th large num is: 7

說明：考慮使用合併排序將輸入陣列排序(時間複雜度為 $O(n\log n)$ )，因排序後就可以直接用 $O(1)$ 定址到第k大的數值。此題必須經過排序，因此最速解時間複雜度應為 $O(n\log n)$ 。

(二)假設陣列為已由小到大排序(不然排序的時間複雜度必大於 $O(n)$ )，考慮以單迴圈(平均時間複雜度為 $O(n)$ )撰

寫，演算法表示如下：

- 1.S為搜尋陣列，n為陣列長度，k為使用者指定搜尋第k大數之條件
- 2.i為陣列索引，初始化為n(指在陣列尾端之後)
- 3.當i不為0 且 k不為1時，i減1，j減1
- 4.重複執行第4步，調整i往陣列前方尋找第k大
- 5.傳回S[i]

八、假設類別A、B、C、D的繼承關係如下：`classA；classB；classC:B；classD:C {A object；}`  
`main() {D objectD；}`，程式執行後，請寫出解構函式的呼叫順序。(5分)

<b>試題評析</b>	由程式語法可推斷本題為C++物件導向題型，而考點在物件合適時會被建立及何時會被釋放，相似於103原住民四等與102年普考計概選擇題。由於C++物件建立的順序是從祖父類別開始建立，但釋放時則是以本身解構子先釋放後再依序往祖父類別釋放，此外C++中物件只要在函數中宣告即會以「堆疊動態變數」方式建立，且離開函數時就會自動被收回，因此亦會增加判斷上難度。
<b>考點命中</b>	《高點·高上程式語言講義》第四回，金乃傑編撰，頁99-100。

**答：**

根據題目為C++語法，依照題意將繼承關係整理如下：

A、B為獨立的類別；

C繼承自B；

D繼承自C；

object為A類別；

主程式中objectD為D類別。

由於題目採用C++語法，故物件在宣告時建立，而解構函數由子類別解構函數先行解構，再逐層往父類別解構，故其順序為：

~D, ~A, ~C, ~B

九、考慮下列文法： $S \rightarrow SS | (S) | ()$  (每小題4分，共12分)

(一)請指出此文法的terminal symbol、non-terminal symbol及start symbol。

(二)針對() $(())$ 字串，推導出left-most derivation sequence。

(三)針對 $((()))()$ 字串，推導出right-most derivation sequence。

<b>試題評析</b>	本題為BNF相關考題，屬於程式語言中較容易拿分的部分，相似於102升官、104地特、104身心障礙考題。考生若能了解BNF中各種節點的定義，並且能熟悉最左推倒與最右推倒之步驟，此題應可獲得高分。
<b>考點命中</b>	《高點·高上程式語言講義》第六回，金乃傑編撰，頁8-13。

**答：**

(一)依照題意說明此文法之符號：

1.terminal symbol: 左括號(、右括號)

2.non-terminal symbol: S

3.start symbol: S

(二)以最左推導法推導() $(())$ 字串之過程如下：

1.S

2.SS

3.()S

4.() $(())$

5.() $(())$ S

6.() $(())$

(三)以最右推導法推導 $((()))()$ 字串之過程如下：

- 1.S
- 2.SS
- 3.S()
- 4.(S)()
- 5.((S))()
- 6.(((O)))()

# 高點 · 高上

【版權所有，重製必究！】