

《程式語言》

試題總評析

相較於往年，今年的題目較為「復古」，除了要寫Java特性外，完全沒有觸及網路程式語言的範圍。另外這次也沒有程式撰寫的題目，而比較、申論題目比例較高，使考生分數可能變得會較為主觀。整體而言，整張考卷題目偏易，程度較好的考生可以展現80分以上的水準；而程度中等的考生也很有機會拿到60分以上的分數。

一、舉出2種影響程式語言設計之重要因素，並各舉一例說明之。(10分)

試題評析	此題屬於程式語言基礎中的題目，已經好幾年沒有考出。如果以評估程式語言或從程式語言的特性切入，仍然可以拿到部分分數。
考點命中	1.《高點程式語言講義第六回》，金乃傑編著，頁60-61。 2.《高點程式語言總複習講義》，金乃傑編著，頁76，考點30。

答：

(一)計算機架構(Computer Architecture)：主要分為依序處理(范紐曼機器，von Neumann Machine)與平行處理(Parallel Machine)兩種機器架構。

1. 早期的程式語言多基於依序處理的架構設計的，在依序處理的系統中，所有的程式與資料都是放在相同機器的記憶體，並且依序送到CPU中處理。例如：C語言即是依照此架構設計的，當成是編譯完成後，會將程式碼與程式中的資料使用相同的位址空間儲存。
2. 在平行處理架構中，程式允許在執行中被插斷，支援多的指令被同時執行，例如：Ada語言，其程式單元為task，task可以同時被處理。

(二)程式設計方法論(programming design methodologies)：隨著時代的演進，硬體成本逐漸降低、程式專案越來越複雜，而產生不同的設計概念，如：非結構化、結構化、由上而下設計、函數型程式設計與物件導向程式設計等。

在非結構化時期，程式語言每一行代表一個工作，程式碼則依序執行；到了結構化與由上而下的時期，程式語言通常使用函數(function)作為程式單元，設計師將相同功能的命令包在函數中，在不同的程式段中呼叫；到了物件導向時期，程式語言支援將程式的資料與處理資料的方法包在類別中，並且透過物件實作類別，達到資料處理的目的。

二、執行下列C++程式，如果有錯誤，寫出所犯之錯誤，如果沒錯誤，寫出所列印之結果：(15分)

C++測試程式：

```
#include "stdafx.h" // VisualC所需要
#include <iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[]) // VisualC所需要之main()
{
    int mdata[] = {5, 2, 8, 0, 10, -8, 3, 7, 18, -4};
    int i = 11, j = 10, k = 0, s = 0, a = 0;
    for (int i = 1; i < j-- - k;)
    {
        if (j > k++ * 3 && (a = mdata [i++]) > 0) s += i++;
        cout<<" i="<<i<<" j="<<j<<" k="<<k<<" s="<<s<<" a="<<a<<endl;
    };
    cout<<"* i="<<i<<" j="<<j<<" k="<<k<<" s="<<s<<" a="<<a<<endl;
    for (int s = 0; i <= 12; i++)
```

高點·高上·特考·gover.sun@gmail.com.tw 台北市開封街一段2號8樓 02-23318268
 【中壢】中壢市中山路100號3樓之3 4256899 【台中】台中市東區復興路四段231-3號1樓 04-22298699
 【台南】台南市中西區中山路147號3樓之1 06-2235868 【高雄】高雄市新興區中山一路308號8樓 07-2358996
 ；【另有板橋·淡水·三峽·林口·羅東·逢甲·東海·中技·彰化·嘉義】

```
cout<<"* i="<<i<<" j="<<j<<" k="<<k<<" s"=<<s<<" a" <<a<<endl;
return ();
}
```

試題評析	此題是以C++出題，考迴圈的執行結果。此題關鍵在迴圈內宣告的變數屬於迴圈區域變數，因此屬於「變數領域」的考題。只要細心作答，判斷出正確答案絕非難事。
考點命中	《高點程式語言講義第三回》，金乃傑編著，頁44-49及上課補充。

答：

輸出為：

i=3 j=9 k=1 s=2 a=2

i=4 j=8 k=2 s=2 a=0

i=6 j=7 k=3 s=7 a=10

* i=11 j=6 k=3 s=7 a=10

* i=13 j=6 k=3 s=7 a=10

註：在迴圈中宣告的int i與int s屬於迴圈的區域變數，因此在迴圈區域中，會使用迴圈中宣告的i和s，但離開迴圈之後，外面的空間就讀取不到這兩者的值。

三、簡釋何謂例外處理（Exception handling），提供例外處理對程式語言有何重大益處？就例外處理程式（Exception handler）之程式結構與繫合例外至例外處理程式（Binding exceptions to handler）兩議題，比較C++與Java兩程式語言之設計。（15分）

試題評析	此題考例外處理的優點及比較C++與Java在例外處理上的特性。這題屬於考古題，且在講義中有比較表格直接命中，對班內學員來說應該能駕輕就熟。
考點命中	1.《高點程式語言講義第三回》，金乃傑編著，頁67、71。 2.《高點程式語言總複習講義》，金乃傑編著，頁22-23，考點10。

答：

(一) 例外處理是當系統偵測到軟體或硬體引發的不正常事件或例外時，會主動執行例外處理程式（exception handler）之程式單元來處理此一例外狀況。提供的優點如下：

1. 增加程式強固性（robust）：避免程式因發生例外而中止執行。
2. 提高程式之可維護性：可以依照錯誤種類動態選擇處理方法。
3. 增加程式可讀性：將一般敘述與錯誤處理指令分離，簡化程式設計。

(二) 茲將C++與Java的例外處理比較如下：

1. 程式結構：

- (1) 相同：使用try-catch語法將可能產生例外的程式碼包在try結構中；可使用多個catch處理不同型態的例外。
- (2) 相異：

	C++	Java
函數拋出例外	在函數中使用throw手動拋出例外給呼叫者。	在函數名稱後以throws Exception宣告，當函數內容發生例外時會直接將例外拋向呼叫者。
提供強制處理敘述	無。	在try-catch結構後加入finally區域，將需要強制執行的程式碼寫在finally中。此段程式不論是否發生例外都會被執行。
例外處理需求	由程式設計師自行決定需要處理的地方。	由程式設計師自行決定需要處理的地方；某些特定敘述一定要用例外處理語法包覆，例如：讀寫檔案。



高點 · 高上 高普 特考

【版權所有，重製必究！】

高點·高上高普特考 goldensun.get.com.tw 台北市開封街一段2號8樓 02-23318268
【中壢】中壢市中山路100號14樓·03-4256899 【台中】台中市東區復興路四段231-3號1樓·04-22298699
【台南】台南市中西區中山路147號3樓之1·06-2235868 【高雄】高雄市新興區中山一路308號8樓·07-2358996
【另有板橋·淡水·三峽·林口·羅東·逢甲·東海·中技·彰化·嘉義】

2. 繫合例外至例外處理程式：

	C++	Java
例外產生	由程式設計師手動以throw命令拋出。	系統自動產生。
例外的種類	以各種資料型態作為預設例外； 程式設計師可自行定義例外類別。	提供例外類別（Exception），程式設計師可再透過繼承撰寫自己的例外。提供作業系統例外、硬體例外。

四、就下列BNF文法，寫出下列敘述（Statement）以最左推導法（Leftmost derivation）之詳細推導過程，以及繪出其剖析樹（Parse tree）。（20分）

BNF: $\langle \text{exp} \rangle \rightarrow \langle \text{id} \rangle + \langle \text{exp} \rangle \mid \langle \text{id} \rangle * \langle \text{exp} \rangle \mid (\langle \text{exp} \rangle) \mid \langle \text{id} \rangle$

$\langle \text{id} \rangle \rightarrow w \mid x \mid y \mid z$

敘述： $w + x * (y + z)$

試題評析	此題如往常一樣考BNF剖析樹，不過這次涉及比較深入的「最左推導」議題，只會剖析樹只能拿到一半分數。
考點命中	1. 《高點程式語言講義第六回》，金乃傑編著，頁6-13及上課補充。 2. 《高點程式語言總複習講義》，金乃傑編著，頁68，考點26。

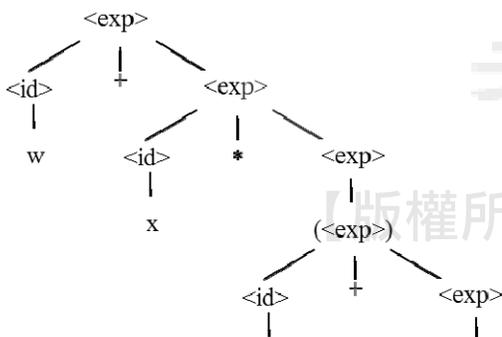
答：

(一)最左推導如下：

$\langle \text{exp} \rangle$
 $\langle \text{id} \rangle + \langle \text{exp} \rangle$
 $w + \langle \text{exp} \rangle$
 $w + \langle \text{id} \rangle * \langle \text{exp} \rangle$
 $w + x * \langle \text{exp} \rangle$
 $w + x * (\langle \text{exp} \rangle)$
 $w + x * (\langle \text{id} \rangle + \langle \text{exp} \rangle)$
 $w + x * (y + \langle \text{exp} \rangle)$
 $w + x * (y + \langle \text{id} \rangle)$
 $w + x * (y + z)$

註：最左推導就是將敘述由左而右分解；反之由最右而左分解則為最右推導。若一個敘述無法由最左或最右分解，則稱為非最左推導或非最右推導。

(二)剖析樹如下：



【版權所有，重製必究！】

五、何謂動態繫合 (Dynamic binding)？比較C++與Java對動態繫合之設計。(20分)

試題評析	此題是動態繫結，物件導向中重要的概念，屬於考古題。不過一樣著重在C++與Java的比較，如果對語法細節不瞭解可能無法得到高分。
考點命中	1.《高點程式語言講義第四回》，金乃傑編著，頁18-19、29。 2.《高點程式語言總複習講義》，金乃傑編著，頁41，考點18、頁49，考點20。

答：

(一)動態繫合又稱動態配置 (dynamic dispatching)，是執行時以動態的方式來決定程序呼叫 (經由訊息傳遞) 與程序定義之繫合。其目的在達成相同的程式碼在執行時期可具有多種執行模式的多模 (Polymorphism)。動態繫合在程式設計中，有以下特點：

- 1.執行彈性高：可以在執行時動態判斷要選用的方法，達到在迴圈中對陣列中對物件操作，可以依照物件實際內容，選用不同的方法。
- 2.減少程式判斷與重複撰寫：由於不必手動判斷每個物件需要執行的操作，因此可以減少程式複雜度。
- 3.可讀性降低：無法一目了然看出物件真正執行的方法，必須要透過實際追蹤，因此可能造成開發時誤判的情形。
- 4.效能低：在執行時才決定呼叫的操作，因此會增加執行的處理時間。

(二)C++與Java在動態繫合的比較：

- 1.程式語法：以程式最後的迴圈輸出形狀的面積，迴圈中語法都相同，都呼叫物件的area方法，但卻因物件實際的內容不同，而執行不同的面積計算，故為動態繫合。

(1) C++寫法：

```

4 class Shape{
5     public:
6         int hight, width;
7         virtual double area(){return 100;};
8 };
9
10 class Rectangle: public Shape{
11     public:
12         double area(){return hight*width; };
13         Rectangle(int h, int w){hight = h; width = w;};
14 };
15
16 class Triangle: public Shape{
17     public:
18         double area(){return (hight*width)/2.0; };
19         Triangle(int h, int w){hight = h; width = w;};
20 };
21
22 int main(){
23     Shape *s[10];
24     for(int i = 0; i < 10; i++){
25         if(i < 5) s[i] = new Rectangle(i+3, 3); //前五個是方形
26         else s[i] = new Triangle(i+3, 3); //後五個是三角形
27     }
28     for(int i = 0; i < 10; i++){ //用同樣的語法輸出
29         printf("s[%d] 面積: %f\n", i, s[i]->area());
30     }
31 }

```

(2) Java寫法：

```

3  class Shape{
4      public int height, width;
5      public double area(){return 100;};
6  }
7
8  class Rectangle extends Shape{
9      public double area(){ return height*width;};
10     public Rectangle(int h, int w){height = h; width = w;};
11 }
12
13 class Triangle extends Shape{
14     public double area(){return (height*width)/2.0;};
15     public Triangle(int h, int w){height = h; width = w;};
16 }
17
18 public class MAIN{
19     public static void main(String[] args){
20         Shape[] s = new Shape[10];
21         for(int i = 0; i < 10; i++){
22             if(i < 5) s[i] = new Rectangle(i+3, 3); //前5個是方形
23             else s[i] = new Triangle(i+3, 3);      //後5個是三角形
24         }
25         for(int i = 0; i < 10; i++){
26             System.out.println("s["+i+"] 面積: "+s[i].area());
27         }
28     }
29 }

```

2.動態繫合使用方法比較：

	C++	Java
使用動態繫合	在函數名稱前加上virtual關鍵字，將函數定義為虛擬函數。	預設為虛擬函數，可以直接執行動態繫合。
不使用動態繫合	預設不使用。	在方法前加上final關鍵字，關閉虛擬函數功能。

由於動態繫合對系統效能的影響很大，且C++中特別強調效能，因此在C++中才預設沒有提供，必須要加上virtual關鍵字才可以使用；但Java主要就是針對物件導向設計的，因此預設提供動態繫合，必須要透過final停止此功能。

【版權所有，重製必究！】

六、瞭解「結構化查詢語言」(SQL: Structural Query Language)是開發網際網路資料庫系統必備的技能，假設一個資料庫系統中有三個資料表，資料表的實體關係圖(Entity-Relationship Diagram)如下圖所示，其中PK表示為主索引鍵，FK表示為外來鍵：



請依序寫出下列各種查詢對應的SQL敘述：(20分)

- (一) 請列出姓名為「甲乙科技」的客戶所有資料。
- (二) 請列出所有姓「王」的業務員姓名。
- (三) 請列出訂單金額曾大於一百萬的客戶名稱。
- (四) 請列出有經手過「甲乙科技」這位客戶訂單的所有業務員的編號與姓名(業務員的資料不可重複列出)。

試題評析	此題屬於資料庫考題，是天外飛來SQL的考題，內容不難，如果對SQL語法稍微熟悉的考生，是「小菜一疊」的送分題。
考點命中	1. 《高點程式語言講義第四回》，金乃傑編著，頁18-19、29。 2. 《高點程式語言總複習講義》，金乃傑編著，頁41，考點18、頁49，考點20。

答：

- (一) SELECT * FROM 客戶 WHERE 名稱 = "甲乙科技" ;
- (二) SELECT 姓名 FROM 業務員 WHERE 姓名 LIKE "王%" ;
- (三) SELECT c.名稱 FROM 客戶編號 c, 訂單 o
WHERE
o.金額 > 1000,000 AND
o.客戶編號 = c.客戶編號
- (四) SELECT DISTINCT 姓名 FROM 業務員 s, 訂單編號 o, 客戶編號 c
WHERE
c.名稱 = "甲乙科技" AND
c.客戶編號 = o.客戶編號 AND
o.業務員編號 = s.業務員編號

【版權所有，重製必究！】