

# 《程式語言》

## 試題評析

今年檢事官題目與往年命題方式相近，基本題目也都占大部分。但考出了程式語言分類等較冷門的試題，且花了20%在了解考生對Cookie概念的熟悉程度，是與以往差異較大之處。另外二、三、五題，都是以往考古題中有類似題的，唯第五題程式設計，因為要讀取多個行數，可能對考生較具挑戰。

總體而言，這次考試難易度偏易，程度好的考生能有80分以上的水準，而程度中等的考生也應能展現60分以上的成績。

一、高階程式語言依其設計理念與執行方式而言，可分成那四大範疇（Paradigm）？(20分)

### 【擬答】

高階程式語言依照設計理念與執行方式，可以分為：命令型語言、函數型式語言、邏輯型式語言與物件導向語言。特性及內容分述如下：

(一)命令型語言（Imperative Language）：

由一連串順序性的敘述所組成，主要概念是透過指定運算改變變數的狀態以完成所要執行的工作，故稱為「命令式語言」。另外，再重複任務上，透過迴圈完成。由於程式設計師必須用敘述描述問題的解決步驟（How to do），故也屬於程序式語言（Procedural Language）。主要實作的語言包括：Pascal、C、Basic、Fortran、COBOL等。

(二)函數型語言（Functional Language）：

由函數呼叫指令所形成的運算式所組成。運算式由函數呼叫指令所構成，成為程式的基本執行單位。在函數型語言中，變數的設定必須經由函數傳入參數來處理，不使用指定敘述；而控制與重複結構也都由特殊的函數呼叫完成。函數型語言還有一個特性是參考透明性（Referential Transparency），指的是函數給予相同的參數，就會有相同的結果，不具有邊際效應。此外函數型語言跟命令型語言一樣，必須要描述問題的解決步驟，故也屬於程序式語言。主要實作的語言包括：LISP、APL、SNOBOL等。

(三)邏輯型語言（Logical Language）：

由一群宣告所組成。宣告是一種符號邏輯（symbolic logic），是一些推理的事實或規則，而不是指定敘述或控制流程，其主要目的為推論（inference），為人工智慧的高階語言。相較於命令型語言，邏輯語言的算術運算效率較低，專長為邏輯符號的處理。在邏輯型語言中，我們不需要告訴電腦要「怎麼做」，只要描述「做什麼（What to do）」，故屬於宣告式程式語言（declarative language）或非程序式語言（non-procedural language）。主要實作為Prolog語言。

(四)物件導向語言（Object-oriented Language）：

由類別（class）實作的物件（objects）所構成，物件間透過訊息（message）溝通操作。有三大特色：

- 1.資料抽象化（Data Abstraction）：將資料及操作封裝（Encapsulation）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.動態繫結（Dynamic Binding）：執行時期動態的決定一個副程式呼叫所要執行的副程式，達成多型（Polymorphism）。

實際上在當其他類型的語言包括物件導向的特性，就可以成為物件導向語言，主要實作語言有：C++、Java、C#、Smalltalk、Ada等。

### 【考點命中】

《高點程式語言講義第六回》，金乃傑編撰，頁57-58。

二、在JAVA語言中，資料成員分成類別資料（class data）與實體資料（instance data）兩大類。請問這兩類資料成員如何宣告？其最主要的區別為何？(20分)

## 【擬答】

類別資料又稱類別變數，是在物件產生前已可使用，此類別的物件可共享交換的資料。

實體資料則是一般的變數，必須要在物件被建立（用new）產生後才會配置空間，也才能被使用的資料。

假設我們宣告一個整數變數a，資料成員宣告方法如下：

- 類別資料：static int a;
- 實體資料：int a;

其宣告的最大差別就在於類別資料使用static關鍵字，以static宣告的變數會在程式載入記憶體時就先配置到Data Segment中，不用等到物件初始化時才安排到Heap Segment中，故在物件還沒被建立時就可以使用。

類別資料還有一個很大的特性是可以達成物件間資料的共享。如前所述，類別資料是放在記憶體中固定的區塊，不會隨著物件釋放而清除，因此可以讓同一類別產生出的不同物件讀取到同一份資料，作為物件間溝通的橋樑。

## 【考點命中】

《高點程式語言講義第四回》，金乃傑編撰，頁28-29。

三、在C#語言中，運算子and then與一般邏輯運算子and的運算功能有何不同之處？(20分)

## 【擬答】

在該語言設計中，利用不同的運算子代表不同之計算。

- and then與or else：捷徑計算(short circuit evaluation)。
- and與or：完全計算(complete circuit evaluation)。

捷徑運算與完全運算比較如下：

(一)捷徑計算的意義：運算式的結果無須計算所有的運算元或運算子，即可決定計算方式。例如：A && B時，若A已經為false，就不會再檢查B，便傳回false。

- 1.優點：計算效率佳、可避免副作用。
- 2.缺點：較易使程式設計師犯下難以除錯之邏輯錯誤。

(二)完全計算的意義：對運算式作求值動作時，必需做完整個運算式，可得出最後的結果。例如：A && B時，若A已經為false，仍會繼續檢查B，但結果一樣是false。

- 1.優點：較不容易使程式設計師犯下難以除錯之邏輯錯誤。
- 2.缺點：計算效率差。

按：在C#語言中沒有and then運算子，題目所指的應為Ada語言，且應該更正為and then運算子（中間有空格）。

## 【考點命中】

《高點程式語言講義第一回》，金乃傑編撰，頁49-50。

四、何謂Cookies？網際網路程式設計為何要使用Cookies？使用Cookies可能會遇到那些問題？(20分)

## 【擬答】

(一)Cookie是網景公司（Netscape）在1993年發明，提供網站辨別使用者身分或儲存使用者瀏覽紀錄資料而儲存在用戶端（Client Side）上的小型文字檔案。一般而言依照保存期限，Cookie可分為持久與非持久兩種。所謂持久Cookie是儲存在使用者硬碟中，除非是過期或手動刪除，不然每次瀏覽網站時都會存取到；而非持久Cookie通常放在記憶體中，當瀏覽器關閉後就會消失。

(二)使用Cookie主要的原因是由於HTTP協議不具有「狀態」，亦即伺服器無法得知使用者上次的操作，致使Web應用程式無法順利執行。常見的Cookie功能如下：

- 1.登入憑證：若使用者成功登入，則將特定的Cookie欄位賦予辨識值，以確保在不同網頁中可以不必要重新登入，卻能夠使用特定權限。另外有些Cookie會提供「永久登入」的功能，讓使用者就算下次連上網站，也可以不用再重新輸入帳號密碼，便能使用，如Gmail即是一例。
- 2.購物車內容：由於在使用者尚未結帳前，寫入資料庫會花費許多額外的成本，因此一般購物網站會透過

Cookie記錄使用者購買的商品品項，以方便結帳時一次列出商品內容。

3. 瀏覽追蹤：與購物車內容相似，但是列出使用者看過那些產品。這一樣是常用在購物網站的技術，例如：雅虎拍賣就會在右側列出使用者最近瀏覽過的產品資料，以方便使用者反覆比較。
4. 偏好設定：許多網站會讓使用者提供客製化的設定，比如語系的選項、版面配色、呈現主題等，這些設定選項都可以透過Cookie完成。網頁應用程式可以在讀取Cookie後呈現針對使用者客製化的網站，提升使用者滿意度。

(三)由於Cookie會記錄使用者的瀏覽資訊，因此可能會發生以下問題：

1. 權限遭他人冒用：若使用者在「永久登入」的網站上忘記登出，則其他使用者連到該網站就可以冒用使用者身分進行操作，甚至詐騙使用者的朋友。這種問題通常在公共場所的電腦中發生。
2. 隱私與安全性：Cookie記錄使用者的行為，如不當使用可能可以整合其他網站、其他方式蒐集到的使用者資訊，比如包含使用者常用的帳號、電腦IP、曾經瀏覽的內容及問卷調查的資料等，建立使用者的剖繪（Profile），侵犯使用者隱私權。
3. Cookie盜賊：駭客可以透過技術蒐集使用者的其他Cookie，侵犯使用者個人隱私或冒用使用者身分操作網站。

【考點命中】

《高點資訊管理與資通安全講義第三回》，金乃傑編撰，頁15-16。

五、請用您熟習的語言，設計一程式，用來計算輸入資料的字元數及行數。(20分)

【擬答】

用C語言，假設輸入資料為一文字檔案，則程式撰寫如下：

```

3  int main(){
4      FILE *fp = fopen("test.txt", "r"); //假設輸入的資料為檔案
5      char line[80];
6      int word_count = 0, line_count = 0;
7
8      while(fgets(line, sizeof(line), fp)){
9          int length = 0; char *s=line;
10         while(*s++ != '\0') length++; //長度算到句尾(\0)
11
12         word_count+=length-1; // -1為扣掉換行符號
13         line_count++;
14     }
15     fclose(fp);
16
17     printf("字元數: %d, 行數: %d", word_count+1, line_count);
18     //最後一行沒有換行，再加回來
19 }
```

【考點命中】

《高點程式語言講義第三回》，金乃傑編撰，頁13-14。