

《程式語言》

一、請說明下列之變數生命期 (Variable lifetime)，並利用 C 語言舉例說明：

(一)Explicit heap-dynamic (5 分)

(二)Static (5 分)

(三)Stack-dynamic (5 分)

命題意旨	本題在測驗考生對於不同記憶體配置之變數生命週期的掌握，並了解學生是否能以 C 程式語言表達不同記憶體變數。
答題關鍵	本題依題目順序作答，應先寫出該種記憶體配置變數之生命期起始與結束之時間點（或事件），再以程式輔助舉例之。
考點命中	《高點程式語言講義》第三回，金乃傑編撰，頁 69-70。

【擬答】

(一)Explicit heap-dynamic：外顯堆積動態變數

1.生命期開始：執行時期，進行動態記憶體配置語法時。

2.生命期結束：程式結束或執行手動釋放物件時。

3.C 語言例子：

```
int * ptr = new int[10]; //建立外顯堆積動態變數，生命期開始
```

```
...
```

```
delete ptr; //手動釋放記憶體，生命期結束
```

(二)Static：靜態變數

1.生命期開始：程式載入時，隨行程（process）配置記憶體，儲存在行程中 Data 區段中，若使用者沒有初始值，自動初始化為 0。

2.生命期結束：程式結束時，隨行程歸還記憶體空間。

3.C 語言例子：

```
void fun1(){
```

```
    static int a;
```

```
    //宣告靜態變數，
```

```
    //但其實生命期已在進入程式前開始
```

```
}
```

(三)Stack-dynamic：堆疊動態變數

1.生命期開始：進入副程式，配置活動紀錄時建立，建立在活動紀錄中。

2.生命期結束：副程式結束，釋放活動紀錄時釋放。

3.C 語言例子：

```
func2(){
```

```
    int a = 5;
```

```
    //宣告堆疊動態變數，
```

```
    //進入副程式時配置記憶體空間
```

```
}
```

二、假設有一個三維矩陣 (3 dimensional matrix)，三個維度分別是 $(2M+1)$ 、 $(2M+1)$ 、 $(2M+1)$ 。

每一個元素 (element) 所占用的記憶體單元為 N bytes。若矩陣的第一個元素之位址為 K ，請回答下列問題：

(一)可否找出三維矩陣中心點之元素位址？若可，請列出計算公式。(10 分)

(二)在(一)中之公式，是否會因程式語言之不同而異，請說明之。(5 分)

命題意旨	本題在測驗考生是否能進行陣列的位址計算，要求考生寫出特定位置陣列元素之記憶體位址。第二小題根據第一小題之答案延伸，測驗考生是否了解不同程式語言之陣列配置方式可能不同（分為列優先及行優先），並進一步考驗考生是否了解不同方法之實際排序邏輯。
------	--

答題關鍵	考生需掌握陣列記憶體位址就是特定元素依期排序方法「前面有多少元素」之概念，即可計算出中心點位址前面有幾行（或列），完成公式；並回答出不因陣列配置方式不同而中心點位置前之元素數仍不變。
考點命中	《高點程式語言講義》第一回，金乃傑編撰，頁 19。

【擬答】

(一)三維陣列之中心點公式如下：

$$K + N * (2M+1) * (2M+1) * M + 2M+1 * M + M$$

(二)不會，雖然語言不同陣列方式可以是列優先或行優先，但因中心點位置不論從列算還是從行算，必須經過的距離相等，因此不論使用何種語言都能得到相同結果。

三、試述程式語言中綁定（binding）的意義，並說明可能的 binding time，也就是 binding 發生的幾個時間點。（20 分）

命題意旨	本題在測驗考生是否能掌握四種基本的繫結時間。
答題關鍵	說明四種基本的繫結時間之內容，並搭配舉例輔助。
考點命中	《高點程式語言講義》第三回，金乃傑編撰，頁 62。

【擬答】

繫結（Binding）指程式實體（program entity）與其屬性（attributes）發生關聯（association）的動作稱之，依照不同時間點，可分為四個時期，說明如下：

時期	內容	例子
定義 (Definition)	限制程式語言所提供的資料型態、符號、變數名稱寫法、函數…的功能與意義。	while 具有重複執行的功能。 C 語言陣列只有一維。 等號(=)具有指派變數的作用。
實作 (Implementation)	所有跟機器相關的特性及變數可能的值。在不同計算機中即使相同的數學運算式，亦可能得到不同的結果；固定在某種機器上才有的特性，另一台機器可能就無法執行。	在 32 位元系統整數範圍為-214783648 至 214783647。
編譯 (Compilation / Translation)	由程式設計師定義資料變數與型態的關係。	宣告 int a，則將變數 a 繫結到整數型態。
	由編譯器定義變數所使用的操作方法（overload）。	x = x+10，「+」執行何種型態的相加，由編譯器決定。
	連結器連結外部二進位檔案程式。	extern int e 與外部.obj 檔案中的變數 e 繫結。
執行 (Execute)	載入時期，將靜態變數、全域變數繫結到指定的記憶體位址。	配置全域變數。
	進入副程式時，在活動紀錄中配置區域變數空間。	foo(){ int x } 則呼叫 foo()時，系統在活動紀錄配置 x 變數的空間。
	執行動態記憶體配置語法，在 Heap 中配置物件空間。	Obj o1 = new Obj(); 執行此敘述時，配置 o1 物件空間。
	執行指定敘述時，將新的值與變數名稱產生關聯。	執行到 i = i + 3 敘述時，將 i 與新的變數值繫結。

四、試說明 C++ 程式語言中的異常處理（Exception Handling）之形式，也就是它的構造（construct）。在一些沒有異常處理設計之語言中，可以做異常處理嗎？亦說明之。（15 分）

命題意旨	本題在測驗考生對 C++ 例外處理之熟悉程度，以及例外處理之運作方式、功能與優點。
答題關鍵	第一小題應先解釋例外處理之定義，說明 C++ 之例外處理程式 try...catch 結構，並以程式碼舉例。 第二小題關鍵在例外處理的觸發方式在無例外處理敘述的語言中是否可以觸發，當觸發後是否可以透過程式進行「防止程式無法執行」的處理，若能，即表示可做例外處理。
考點命中	《高點程式語言講義》第三回，金乃傑編撰，頁 94-95。

【擬答】

(一)例外處理適當系統偵測到軟體或硬體引發無法執行正常指令的事件所進行的程式單元 (exception handler)，使程式回復到可以繼續執行的狀態稱之。在 C++ 的例外處理中有以下構造：

1. 使用 try{} 包住可能出現例外的程式碼。
2. 在產生無法正常執行的敘述使用 throw 關鍵字將例外拋出。在 C++ 中例外是以資料型態或變數來代表，如可用 throw int 拋出一個整束型態的例外。
3. 使用 catch(){} 包住處理例外的敘述，其中 catch 的括號內為相對應的資料型態，亦可使用... 表示可處理任何型態的例外。

考慮 C++ 例外處理程式如下：

```

3  int main(){
4      int i = 0, arr[] = {2, 3, 5, 7, 11};
5      try{
6          while(1){
7              if(i > (sizeof(arr)/sizeof(int)-1)) throw i;
8              printf("%d\n", arr[i++]);
9          }
10     }catch(int){
11         printf("Caught Integer exception!!!");
12     }
13 }
```

(二)在無異常處理的語言中，仍可達到部分異常處理的功能。由於異常處理指的是一種讓程式在發生異常時回復到可正常執行的狀態，而造成程式發生異常的原因很多，若為資料所引起的，例如算術運算中「分母是 0」的問題，在無異常處理的語言中，可以透過 if 判斷式檢查出來，顯示錯誤訊息，跳過該算式，即可達到異常處理的功能。但若程式中的異常是硬體所引起，如記憶體不足，則無法透過程式碼檢測出來，故無法進行異常處理。另外，異常處理的一個重要特性是將處理正常邏輯與異常邏輯的程式碼分離，達到可讀性、可維護性，但若該語言中無異常處理敘述（如 try...catch），則無法達到該特性。

五、程式執行過程常會產生啟動記錄 (activation record)，請解釋什麼是 activation record。若有一 C 副程式，其參數 (parameter) 與區域變數 (local variables) 如下，請以圖示說明其 activation record。(20 分)

```

void fun1(int x, float y)
{ float    data[3];
  Int      count;
  float    total;
  ...
}
```

命題意旨	本題在測驗考生對於活動紀錄結構的了解，並需對照真實程式完成活動紀錄之格式內容。
答題關鍵	考生需掌握副程式中的形式參數儲存在活動紀錄的參數欄位，而副程式中宣告的變數若前無 static 關鍵字則除存在活動紀錄中區域變數欄位。此外該題中對於程式提供的資訊有限，僅有區域變數與參數部分，故需自行假設其他資訊值才能完整說明活動紀錄內容。
考點命中	《高點程式語言講義》第二回，金乃傑編撰，頁 65。

【擬答】

啓動紀錄 (Activation Record) 是電腦中的記憶體區塊，儲存副程式執行使用的資料。副程式被呼叫時系統為副程式建立，當副程式結束後該資料結構即被釋放。假設題目中的 fun1 函數被 main() 所呼叫，則以圖示說明啓動紀錄欄位資料如下：

欄位名稱	值
Return Value	未知
Return Address	main() 中呼叫到 fun1() 的程式位址
Dynamic Link	main()
Static Link	NULL
Local Variable: float data[0]	
Local Variable: float data[1]	
Local Variable: float data[2]	
Local Variable: int count	
Local Variable: float total	
Parameter: int x	
Parameter: float y	

六、給予一個 grammar for expression 如下：

$\langle \text{assignment} \rangle \rightarrow \langle \text{id} \rangle = \langle \text{expr} \rangle$

$\langle \text{id} \rangle \rightarrow X | Y | Z$

$\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{term} \rangle \mid \langle \text{term} \rangle$

$\langle \text{term} \rangle \rightarrow \langle \text{term} \rangle * \langle \text{factor} \rangle \mid \langle \text{factor} \rangle$

$\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle) \mid \langle \text{id} \rangle$

(一) 試列出 $X = Y + Z * X$ 之剖析樹 (parse tree)。(10 分)

(二) 該剖析樹是唯一嗎？(5 分)

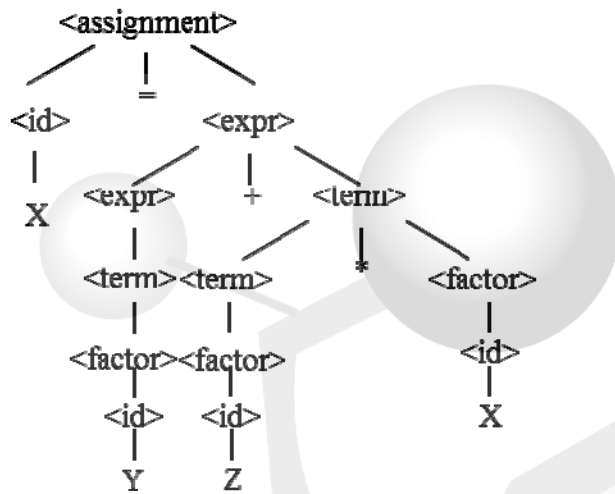
命題意旨	本題在測驗考生對於 BNF 文法轉換為剖析樹之能力，並測驗考生是否能判斷模糊文法之特性。
答題關鍵	<p>考生應根據題目中所提供之文法進行最左或最右之推導，以畫出剖析樹。而判斷式剖析樹是否唯一之判斷點有二：</p> <ol style="list-style-type: none"> 一文法產生規則右邊是否存在相同的符號。 一文法產生規則的結果是否可由其他產生規則組合而成。 <p>若是，則可能存在多棵剖析樹。</p>
考點命中	《高點程式語言講義》第六回，金乃傑編撰，頁 8-9、26。

【高點法律專班】

版權所有，重製必究！

【擬答】

(一)語句 $X = Y + Z * X$ 之剖析樹繪製如下：



(二)該剖析樹是唯一剖析樹，因為在該文法中，產生規則右邊的每個元素都不重複如<expr>可產生<expr>+<term>或<term>，其中每個產生的結果沒有任何相同的符號；且存在以組合產生規則的方式形成其他產生規則。

【高點法律專班】

版權所有，重製必究！