

高點

堅持夢想
全力相挺

公職 EXPRESS 快速通關

Pass!

地特准考證 就是你的 **VIP券**

弱科健檢 

加入【高點·高上生活圈】可免費預約參加 ▶▶▶



113/12/7-31 前 **商管** **會計** **資訊** **地政** 享考場獨家優惠!

114 高普考 衝刺

- 【總複習】面授/網院：特價 4,000 元起、雲端：特價 5,000 元起
- 【申論寫作正解班】面授/網院：特價 3,000 元起科、雲端：特價 7 折起/科
- 【經典題庫班】面授/網院：特價 2,500 元起/科、雲端單科：特價 7 折起
- 【狂作題班】面授：特價 5,000 元起/科

114、115 高普考 達陣

- 【面授/網院全修班】特價 34,000 元起
 - 114年度：再優 10,000 元(高考法制、公職社工師除外，輔限至114.7.31止)
 - 115年度：享 ①再折 2,000 元 + ②線上課程 1 科 + ③ 60 堂補課券 舊生再優 1,000 元
- 【考取班】高考：特價 65,000 元、普考：特價 55,000 元(限面授/網院)

114國營

- 【企管/政風/地政/資訊/財會】
網院全修：特價 25,000 元起、雲端：特價 31,000 元起

單科 加強方案

- 【114年度】面授/網院：定價 65 折起、雲端：定價 85 折
舊生贈圖禮：500 元

※優惠詳情依各分班櫃檯公告為準

《程式設計概要》

- 一、(一) 請針對C++、Java 及Python 三種程式語言，以列表方式比較四種特性：執行方式（編譯或直譯）、物件導向繼承機制、性能（執行速度）、應用場景領域。（15分）
- (二) 請用C、C#、C++、Java 或Python程式語言撰寫一個程式，定義一個100元素的陣列，每一個元素可以儲存的資料包含10個字元長度的字串，用以儲存姓名，一個整數用以儲存薪資，及一個浮點數用以儲存年資，提供使用者一筆一筆輸入這些資料。（15分）

試題評析	本考題能有效測試考生的程式語言基礎及應用能力。第一題要求比較 C++、Java 和 Python 三種語言的特性，能檢視考生對語言的執行方式、物件導向特性及應用場景的理解。第二題則測試考生在多種語言中運用陣列和資料結構處理實際問題的能力，並檢查其邏輯清晰度和程式撰寫熟練度。這兩題的評分權重平均分配，能公平地評估理論知識與實作能力，用功的考生可以獲得高分。
考點命中	1.《高點·高上程式設計概要講義》第二回，許振明編撰，頁120-121。 2.《高點·高上程式設計概要講義》第二回，許振明編撰，頁185。 3.《高點·高上程式設計概要講義》第二回，許振明編撰，頁235。

答：

(一) C++、Java與 Python的比較

特性	C++	Java	Python
執行方式	編譯語言：需將程式碼編譯為機器碼後執行，速度快。	半編譯半直譯：先編譯為位元組碼，透過JVM 解譯執行。	直譯語言：程式碼逐行解譯執行，開發速度快，但效能較低。
物件導向繼承機制	支援多重繼承，透過虛擬函數解決多重繼承的衝突問題。	單一繼承(使用介面實現多型)，不支援多重繼承。	支援多重繼承。
性能（執行速度）	極高效能，接近底層硬體，適合高性能應用。	效能介於 C++ 和 Python之間，執行效率較穩定。	效能較低，但適合以開發速度為優先的應用場景。
應用場景領域	系統程式設計、遊戲開發、嵌入式系統、驅動程式等高性能需求領域。	企業應用程式、跨平台系統、Android 應用程式開發。	資料分析、機器學習、快速原型開發、Web 開發。

(二) C語言程式

```
#include <stdio.h>
#include <string.h>
```

```
// 定義結構用於儲存每個人的資料
```

```
struct Employee {
    char name[11];        // 姓名 (10 個字元 + '\0')
    int salary;          // 薪資
    float years_of_service; // 年資
};
```

```
int main() {
    struct Employee employees[100]; // 定義 100 個元素的陣列
```

```
int count = 0;           // 記錄已輸入的人數
char choice;

printf("歡迎使用員工資料輸入系統！\n");

// 提供使用者逐筆輸入資料
do {
    if (count >= 100) {
        printf("已達到最大輸入數量（100）！\n");
        break;
    }

    printf("\n輸入第 %d 筆資料：\n", count + 1);

    // 輸入姓名
    printf("請輸入姓名（最多 10 個字元）：");
    scanf("%10s", employees[count].name); // 限制最多輸入 10 個字元

    // 輸入薪資
    printf("請輸入薪資：");
    scanf("%d", &employees[count].salary);

    // 輸入年資
    printf("請輸入年資（小數點格式）：");
    scanf("%f", &employees[count].years_of_service);

    count++; // 計算輸入筆數

    // 詢問是否繼續輸入
    printf("是否繼續輸入資料？(y/n)：");
    scanf(" %c", &choice);
} while (choice == 'y' || choice == 'Y');

// 顯示已輸入的所有資料
printf("\n輸入的員工資料如下：\n");
for (int i = 0; i < count; i++) {
    printf("第 %d 筆資料：\n", i + 1);
    printf("姓名：%s\n", employees[i].name);
    printf("薪資：%d\n", employees[i].salary);
    printf("年資：%.2f 年\n", employees[i].years_of_service);
}

return 0;
}
```

【版權所有，重製必究！】

程式說明：

1. 結構定義：
使用 struct Employee 定義包含姓名、薪資和年資的結構體。
2. 陣列初始化：

定義 struct Employee employees[100] 作為存儲空間。

3. 輸入流程：

- (1)使用 do-while 迴圈讓使用者逐筆輸入資料。
- (2)限制姓名的輸入長度為 10 個字元，超出部分會自動截斷。
- (3)使用者可以選擇是否繼續輸入資料。

4. 輸出結果：

最後輸出所有已輸入的資料，顯示姓名、薪資和年資。

二、(一) 請將下列C++語言的迴圈程式以for 迴圈方式改寫，並寫出此程式片段輸出的結果。
(5分)

```
int i=1;
while (i <= 10)
{
    if (i < 5 && i !=2)
        std::cout << 'X';
    i ++;
}
```

(二) 下列C++程式有一行錯誤，使得編譯無法通過。請列出那一行錯誤，並寫出正確程式碼。修正後執行此程式，使用者輸入153 後輸出為何？(5分)

1	#include <iostream>
2	using namespace std;
3	int main()
4	{
5	int number, originalNumber, remainder, result = 0;
6	cout <<"請輸入三位數:";
7	cin >> number;
8	originalNumber = number;
9	while (originalNumber != 0)
10	{
11	remainder = originalNumber%10;
12	result += remainder*remainder*remainder
13	originalNumber /= 10;
14	}
15	cout << "輸出為："<< result;
16	return 0;
17	}

(三) 執行下列C++程式，當使用者輸入32，其輸出結果為何？(5分)

```
#include <iostream>
using namespace std;
int main()
{
    int x;
    cout << "請輸入一個正整數:";
```

```

cin >> x;
for (int i = 1; x > 0; i++)
{
    if (x % i == 0)
        cout << i << " ";
    if (x == i)
    {
        i = 0;
        cout << "\n";
        cout << "請輸入一個正整數：";
        cin >> x;
    }
}
return 0;
}

```

(四) 請利用C, C#, C++, Java或Python程式語言寫一個副程式，求二個正整數的最大公因數(GCD)及最小公倍數(LCM)。(10分)

試題評析	本試題測試考生的 C++ 程式設計能力，涵蓋基本語法、邏輯分析與程式撰寫。第一題要求用 for 迴圈改寫，考察考生對迴圈結構的靈活運用。第二題檢查錯誤，測試考生的程式除錯與修正能力。第三題要求輸出分析，評估學生對程式執行流程與條件判斷的理解。第四題則需要考生撰寫實用副程式，應用數學演算法解決問題，測試綜合實作能力。題目難度適中，範圍完整，能有效評估考生的程式設計水平。
考點命中	<ol style="list-style-type: none"> 1. 《高點·高上程式設計概要講義》第一回，許振明編撰，頁44。 2. 《高點·高上程式設計概要講義》第一回，許振明編撰，頁76。 3. 《高點·高上程式設計概要講義》第二回，許振明編撰，頁85-86。

答：

(一)

```

for (int i = 1; i <= 10; i++) {
    if (i < 5 && i != 2) {
        std::cout << 'X';
    }
}

```

程式片段的輸出結果

條件 $i < 5 \ \&\& \ i \neq 2$ 表示當 i 小於 5 且不等於 2 時，輸出 'X'。

迴圈 $i = 1$ 到 $i = 10$ ：

- 當 $i = 1$ ：滿足條件，輸出 X。
- 當 $i = 2$ ：不滿足條件（因為 $i \neq 2$ 為假）。
- 當 $i = 3$ ：滿足條件，輸出 X。
- 當 $i = 4$ ：滿足條件，輸出 X。
- 當 $i = 5$ ：條件 $i < 5$ 不成立，停止輸出。

從 $i=6$ 到 $i=10$ ：條件 $i < 5$ 不成立，不輸出。

最終輸出結果

XXX

(二)

(1)第12列少結束符號

`result += remainder * remainder * remainder;`

(2)

執行結果分析

使用者輸入 153

`originalNumber = 153`，進入迴圈：

`remainder = 153 % 10 = 3`

`result += 3 * 3 * 3 = 27`

`originalNumber /= 10 = 15`

`originalNumber = 15`，繼續迴圈：

`remainder = 15 % 10 = 5`

`result += 5 * 5 * 5 = 125`

`result = 27 + 125 = 152`

`originalNumber /= 10 = 1`

`originalNumber = 1`，繼續迴圈：

`remainder = 1 % 10 = 1`

`result += 1 * 1 * 1 = 1`

`result = 152 + 1 = 153`

`originalNumber /= 10 = 0`

迴圈結束，`result = 153`。

輸出為：153

註：153為三位數的阿姆斯壯數

(三)

程式功能解讀

程式邏輯：

要求使用者輸入一個正整數 x 。

使用 `for` 迴圈，從 $i=1$ 開始遞增，列出所有可以整除 x 的因數。

當 $i == x$ 時：

將 i 重置為 0。

要求再次輸入 x ，重新開始迴圈。

關鍵問題：

條件 `for (int i = 1; x > 0; i++)` 中， $x > 0$ 將導致無限迴圈（只要使用者輸入的值是正數）。

程式中並未限制輸入次數，因此程式會持續要求輸入新的數值。

當使用者輸入 32 時的執行過程

初次輸入 $x = 32$ ：

1.迴圈啟動， i 從 1 開始遞增。

2.判斷條件 $x \% i == 0$ ，列出所有可以整除 32 的 i 值。

$i = 1, 32 \% 1 == 0$ ，輸出 1。
 $i = 2, 32 \% 2 == 0$ ，輸出 2。
 $i = 4, 32 \% 4 == 0$ ，輸出 4。
 $i = 8, 32 \% 8 == 0$ ，輸出 8。
 $i = 16, 32 \% 16 == 0$ ，輸出 16。
 $i = 32, 32 \% 32 == 0$ ，輸出 32。

3.當 $i == 32$ 時，執行：
 重置 $i = 0$ 。
 要求再次輸入 x 。

第二次輸入：
 使用者再次輸入數值（例如 32），程式重複上述步驟。

假設使用者只輸入一次 32 並強制停止：
 輸出為：
 1 2 4 8 16 32

如果使用者不停止，程式將繼續要求輸入新值並重複上述步驟，形成無限迴圈。

(四)

```
#include <stdio.h>
```

```
// 求最大公因數 (GCD) 的副程式
```

```
int gcd(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}
```

```
// 求最小公倍數 (LCM) 的副程式
```

```
int lcm(int a, int b) {
    return (a * b) / gcd(a, b); // 利用 GCD 計算 LCM
}
```

```
int main() {
```

```
    int num1, num2;
```

```
    // 輸入兩個正整數
```

```
    printf("請輸入兩個正整數: ");
    scanf("%d %d", &num1, &num2);
```

```
    // 檢查輸入是否合法
```

```
    if (num1 <= 0 || num2 <= 0) {
        printf("請輸入正整數！\n");
        return 1;
    }
}
```

```

// 計算 GCD 和 LCM
int gcd_result = gcd(num1, num2);
int lcm_result = lcm(num1, num2);

// 輸出結果
printf("最大公因數 (GCD): %d\n", gcd_result);
printf("最小公倍數 (LCM): %d\n", lcm_result);

return 0;
}

```

程式說明

- 輾轉相除法求 GCD：
 - 在 gcd 函數中，使用輾轉相除法計算兩個正整數的最大公因數：
 - 如果 $b == 0$ ，則 a 為最大公因數。
 - 否則，繼續執行 $b = a \% b$ ，直到 $b == 0$ 。
- 計算 LCM：
 - 使用公式：
 $\text{lcm}(a,b) = a \times b / \text{gcd}(a,b)$
 - 這個公式有效避免了重複計算，也確保數值的正確性。
- 主程式 (main)：
 - 要求使用者輸入兩個正整數。
 - 呼叫 gcd 和 lcm 函數計算結果。
 - 輸出 GCD 和 LCM。

三、(一) 下列C++程式，當程式執行後輸入二個正整數，「7 10」，其輸出為何？（5 分）

```

#include <iostream>
using namespace std;
void getNumbers(int& input1, int& input2);
void function1(int& var1, int& var2);
void function2(int var1, int var2);
void showResult(int result1, int result2);
int main()
{
    int first, second;
    getNumbers(first, second);
    function1(first, second);
    showResult(first, second);
    return 0;
}

```

```
void getNumbers(int& input1, int& input2)
```

```

{
    cout <<"輸入二個正整數： ";
    cin >> input1 >> input2;
}
void function1(int& var1, int& var2)
{
    int temp;
    temp = var1;
    var1 = var2;
    var2 = temp;
}
void showResult(int result1, int result2)
{
    cout <<"二個數輸出結果為：" << result1 << " " << result2 << endl;
}

```

(二) 若C++程式包含下列宣告：

```

class Automobile
{
public:
    void setPrice(double newPrice);
    void setProfit(double newProfit);
    double getPrice();
private:
    double price;
    double profit;
    double getProfit();
}

```

且主程式 (main function) 包含下列宣告，並設定完成其member 變數中的值。

```
Automobile Toyota, Nissan;
```

下列那些指令敘述 (statements) 在主程式中是合法的 (legal)，那些是不合法的 (illegal)？(10 分)

```
Toyota.price = 5233.99;
```

```
Nissan.setPrice(4000.24);
```

```
double aPrice, aProfit;
```

```
aPrice = Nissan.getPrice();
```

```
aProfit= Nissan.getProfit();
```

```
aProfit= Toyota.getProfit();
```

```
Toyota = Nissan;
```

- (三) 請說明物件導向程式中多型 (Polymorphism) 的概念。多型通常可以透過多載 (Overloading) 和覆寫 (Overriding) 來實現。請說明多載與覆寫的定義與特性。(15分)

試題評析	本試題考察考生對 C++ 語言和物件導向設計的理解與應用能力。第一題通過執行流程的輸出分析，測試考生對程式邏輯和條件的掌握。第二題聚焦於成員變數的存取合法性，評估考生對語言作用域與存取控制的理解。第三題則深入物件導向的多型概念，要求考生清楚闡述多載與覆寫的特性，檢驗其對設計模式和應用場景的認識。題目層次分明，理論與實作結合，能有效反映考生的程式設計綜合能力，用功的考生可獲得高分。
考點命中	1.《高點·高上程式設計概要講義》第二回，許振明編撰，頁132-133。 2.《高點·高上程式設計概要講義》第二回，許振明編撰，頁191-193。 3.《高點·高上程式設計概要講義》第二回，許振明編撰，頁268-269。

答：

(一)

程式輸出

當使用者輸入 7 10 時，程式的輸出結果為：
二個數輸出結果為：10 7

程式執行分析

1.函式說明：

```
getNumbers(int& input1, int& input2):
```

接受兩個整數參考參數 (input1 和 input2)。

透過輸入 (cin) 讀取用戶輸入的兩個正整數並賦值給 input1 和 input2。

```
function1(int& var1, int& var2):
```

接受兩個整數參考參數 (var1 和 var2)。

交換這兩個整數的值 (透過第三個變數 temp 實現)。

```
showResult(int result1, int result2):
```

接受兩個整數值參數 (result1 和 result2)。

輸出兩個整數的值。

2.main 函式執行流程：

變數宣告：

宣告兩個整數變數 first 和 second。

呼叫 getNumbers 函式：

要求用戶輸入兩個正整數。

假設用戶輸入 7 10，則：**【版權所有，重製必究！】**

```
first = 7
```

```
second = 10
```

呼叫 function1 函式：

交換 first 和 second 的值：

原本 first = 7 和 second = 10

經過交換後：

first = 10

second = 7

呼叫 showResult 函式：

輸出交換後的兩個數值：

(二)

分析與合法性判斷

(1)Toyota.price = 5233.99;

非法 (illegal)：

price 是 private 成員，無法直接存取。

必須透過公有方法 setPrice(double) 來設定價格。

(2)Nissan.setPrice(4000.24);

合法 (legal)：

setPrice(double) 是公有方法，可以用來設定 price 的值。

(3)double aPrice, aProfit;

合法 (legal)：

宣告了兩個局部變數 aPrice 和 aProfit，用於存儲價格和利潤。

(4)aPrice = Nissan.getPrice();

合法 (legal)：

getPrice() 是公有方法，可以用來取得 price 的值。

aProfit = Nissan.getProfit();

(5)非法 (illegal)：

getProfit() 是 private 成員函數，無法從主程式中存取。

aProfit = Toyota.getProfit();

(6)非法 (illegal)：

同上，getProfit() 是 private 成員函數，無法從主程式中存取。

(7)Toyota = Nissan;

合法 (legal)：

預設的賦值運算符對於類別對象是合法的，這裡會將 Nissan 的所有成員值複製到 Toyota 中。

(三)

1.多型 (Polymorphism) 的概念

多型是物件導向程式設計的一個核心特性，允許不同的類別以相同的介面進行操作，進而提高程式的靈活性和可擴展性。多型的本質是「一個介面，多種行為」，主要用於實現程式的可重用性和動態綁定 (Dynamic Binding)。

(1)靜態多型：在編譯期間確定方法的行為，例如多載(Overloading)。

(2)動態多型：在執行期間決定方法的行為，例如覆寫(Overriding)。

2.多載 (Overloading)

(1)定義：多載是指在同一個類別中，方法名稱相同但參數的數量或型別不同。透過多載，可以讓一個方法實現多種功能，具體行為根據輸入參數決定。

(2)特性：

(A)方法名稱相同。

(B)方法的參數數量或型別不同。

(C)與返回值無關（僅憑返回值不能區分多載）。

(D)多載是靜態多型，在編譯時期就決定使用哪個方法。

(E)範例 (C++) :

```
class Calculator {
public:
    int add(int a, int b) {
        return a + b;
    }
    double add(double a, double b) {
        return a + b;
    }
    int add(int a, int b, int c) {
        return a + b + c;
    }
};
```

(F)解釋：

(I)方法名稱皆為 add。

(II)方法參數的數量和型別不同（整數、浮點數、三個參數）。

(III)編譯器根據方法的參數選擇適合的 add 方法。

3.覆寫 (Overriding)

(1)定義：覆寫是指子類別重新定義從父類別繼承而來的方法，方法名稱、參數和返回值完全相同。透過覆寫，子類別可以實現與父類別不同的行為。

(2)特性：

(A)方法名稱相同。

(B)方法參數和返回值型別相同。

(C)必須存在父子類別的繼承關係。

(D)覆寫是動態多型，方法的選擇取決於執行期間的對象型別。

(E)使用 virtual (C++) 或 override 關鍵字（某些語言中）明確表示方法覆寫。

(F)範例 (C++)：

```
class Animal {
public:
    virtual void sound() {
        cout << "Animal makes a sound" << endl;
    }
};
```

```
class Dog : public Animal {
public:
    void sound() override { // 覆寫父類別的方法
        cout << "Dog barks" << endl;
    }
};
```

```
int main() {
    Animal* animal = new Dog();
    animal->sound(); // 動態綁定，呼叫 Dog 的 sound 方法
    delete animal;
}
```

(G)解釋：

(I)父類別 Animal 定義了 sound() 方法。

【版權所有，重製必究！】

(II)子類別 Dog 覆寫了 sound() 方法，提供自己的實現。

(III)使用父類指針指向子類別對象時，執行時期會根據實際對象決定呼叫哪個方法。

四、(一) 請說明下列二個PHP 程式的輸出結果。(8 分)

1.

```
<!DOCTYPE html>
<html>
<body>
<?php
$i = 1;
while ($i < 6) {
    if($i == 3) break;
    echo $i;
    $i++;
}
?>
</body>
</html>
```

2.

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 5;
function myTest() {
    echo "<p>Variable x inside function is: $x</p>";
}
myTest();
echo "<p>Variable x outside function is: $x</p>";
?>
</body>
</html>
```

【版權所有，重製必究！】

(二) 請寫一個PHP程式，隨機產生一個0-100的數。若該數小於等於50，則輸出該數及「小」，若大於50 小於100，則輸出該數及「大」。例如當產生65 時，輸出為「65 大」。(7 分)

試題評析

本試題針對PHP語言設計，著重考生的基礎知識和應用能力。第一題要求分析兩段PHP程式的輸出結果，測試考生對程式執行流程、條件判斷與變數作用域的理解。第二題則要求考生撰寫

	程式，隨機生成數字並根據條件輸出結果，評估其程式邏輯設計與實現能力。題目分值分配合理，難度適中，涵蓋了理論分析與實作兩方面，有助於全面檢驗考生對PHP語言的掌握程度。
考點命中	1. 《高點·高上程式設計概要講義》第二回，許振明編撰，頁64。 2. 《高點·高上程式設計概要講義》第二回，許振明編撰，頁66-67。

答：

(一)

(1)

12

原始程式碼：

```
<?php
$i = 1;
while ($i < 6) {
    if($i == 3) break;
    echo $i;
    $i++;
}
?>
```

程式邏輯：

- (A)初始化變數 $i = 1$ 。
 (B)使用 while 迴圈，當 $i < 6$ 時執行迴圈內的程式碼。
 每次迴圈：
 (C)判斷 i 是否等於 3，如果是，使用break中斷迴圈。
 (D)如果 i 不等於 3，執行 echo i 輸出 i 的值。
 (E)將 $i++$ ，使 i 增加 1。

輸出結果

在迴圈結束前，執行了兩次 echo，因此輸出為：

12

(2)

`<p>Variable x inside function is: </p><p>Variable x outside function is: 5</p>`

原始程式碼：

```
<?php
$x = 5;

function myTest() {
    echo "<p>Variable x inside function is: $x</p>";
}
```

```
myTest();
echo "<p>Variable x outside function is: $x</p>";
?>
```

程式邏輯：

【版權所有，重製必究！】

- (A)宣告全域變數 \$x = 5\$。
- (B)定義一個名為 myTest() 的函數：
- (C)試圖輸出變數 \$x\$ 的值。
- (D)執行函數 myTest()。
- (E)在函數外輸出變數 \$x\$ 的值。

輸出結果

<p>Variable x inside function is: </p><p>Variable x outside function is: 5</p>

(二)

```
<!DOCTYPE html>
<html>
<body>

<?php
// 隨機產生一個 0 到 100 的數字
$randomNumber = rand(0, 100);

// 判斷數字範圍並輸出結果
if ($randomNumber <= 50) {
    echo $randomNumber . " 小";
} elseif ($randomNumber < 100) {
    echo $randomNumber . " 大";
}
?>

</body>
</html>
```

程式說明

隨機數生成：

使用 rand(0, 100) 函數生成一個 0 到 100 的隨機整數。

條件判斷：

如果隨機數小於或等於 50，輸出數字加上「小」。

如果隨機數大於 50 且小於 100，輸出數字加上「大」。

【版權所有，重製必究！】

高點

用一套書連續成功

高普特考 打通關！

2025
最新版



7月高普考

報名：03/11~03/20 考試：07/04~07/08

12月地方特考

報名：09/09~09/18 考試：12/06~12/08

重點整理



解題完全制霸



工具書



113高普考
命中事實



好書+好課
立即嘗鮮



更多套書

歷屆高手聯合推薦，上榜必讀這套！

一般行政



一般民政



人事行政



財稅行政



會計



高點文化事業
publish.get.com.tw



113/12/10-31高普考書籍特惠中
手刀購買，快至高點網路書店