

115年公務人員特種考試關務人員、身心障礙人員考試及
115年國軍上校以上軍官轉任公務人員考試試題

考試別：身心障礙人員考試

等 別：四等考試

類 科：資訊處理

科 目：程式設計概要

考試時間：1小時30分

座號：_____

※注意：(一)禁止使用電子計算器。

(二)不必抄題，作答時請將試題題號及答案依照順序寫在試卷上，於本試題上作答者，不予計分。

(三)本科目得以本國文字或英文作答。

一、請根據下方 C 語言的程式碼片段，回答下列問題：

```
// 方法 (A)：巨集定義
#define CALC(x, y) (x * y + x)

// 方法 (B)：函式定義
int calc(int x, int y) {
    return x * y + x;
}

int main() {
    int a = 2, b = 3;

    //執行方法 (A)
    int resultA = CALC(++a, b + 2);

    a = 2; // 重置 a 的值進行方法 (B)
    int resultB = calc(++a, b + 2);

    return 0;
}
```

(一)請寫出 $CALC(++a, b + 2)$ 巨集展開後的實際運算式，並據此計算 $resultA$ 的值。(5分)

(二)請問方法 (A) 以及方法 (B) 在參數處理方式及實際執行結果上的差異。(10分)

(三)若將方法 (A) 修正為：

```
#define CALC(x, y) ((x) * (y) + (x))
```

以便解決運算優先順序之問題，請說明是否可以得到與方法 (B) 相同的結果？(10分)

二、請根據下方 Java 程式碼，回答下列問題：

```
class Main {
    public static void main(String[] args) {
        int[] myArr = {10, 20};
        modify(myArr);

        for (int i : myArr) {
            System.out.print(i + " ");
        }
    }

    public static void modify(int[] arr) {
        arr[0] = 100;           // 操作 A
        arr = new int[]{1, 2, 3}; // 操作 B
        arr[0] = 999;           // 操作 C
    }
}
```

- (一)請比較一般程式語言中函式參數傳遞機制，傳值 (Pass by value) 與傳參考 (Pass by reference) 的差異。(5分)
- (二)請寫出上方程式碼執行後輸出的結果。(5分)
- (三)請依據上方程式碼的輸出結果，說明 Java 在傳遞陣列參數之機制，其本質是採用傳值還是傳參考？請結合操作 B 以及其後續的指令，解釋為何輸出結果符合該機制。(15分)

三、請根據下方的 Java 程式碼，回答下列問題：

```
class Animal {  
    void eat() {  
        System.out.print("Generic..");  
    }  
}  
  
class Bird extends Animal {  
    void eat() {  
        super.eat();  
        System.out.print("Using beak(喙)");  
    }  
    void fly() {  
        System.out.println("I can fly");  
    }  
}
```

- (一)請說明過載 (Overloading) 以及覆寫 (Overriding) 在 Java 程式語言上的差別。(3分)
- (二)上方程式碼展現了那種特性 (Overloading 或是 Overriding) ? (2分)
- (三)請寫出執行 `new Bird().eat()` 的輸出結果。(5分)
- (四)根據上方程式碼，Bird 類別具備飛行 `fly()` 的方法，當要新增一個 Ostrich(駝鳥，屬於鳥類但卻不會飛行)類別，若採用直接繼承 Bird 類別會遇到什麼問題？(5分)
- (五)根據(四)所面臨的問題，請提出兩種解決該問題的方法。(10分)

四、下方 Java 程式碼中，process() 方法根據支付方式之字串會執行不同邏輯。此設計的問題是：每新增一種支付方式，都必須修改 process() 方法。為了解決這個問題，可以透過繼承與多型的方式提高程式維護性。請使用 Java 的繼承與多型重構此程式，並完成下列要求：

```
public class PaymentService {
    public void process(String type, double amount) {
        if (type.equals("CreditCard")) {
            System.out.println("Processing Credit Card: " + amount);
        } else if (type.equals("PayPal")) {
            System.out.println("Processing PayPal: " + amount);
        }
        // 未來若增加 TaiwanPay，必須在此繼續修改代碼
    }
}
```

- (一) 定義一個抽象類別 Payment。(5 分)
- (二) 將不同支付方式 (包含新增之 TaiwanPay) 的處理邏輯移至對應子類別。(15 分)
- (三) 寫出重構後呼叫端使用方式。(5 分)