

大吉

總複習班 → 提升統整力

- 求勝科目** 共同科目+專業科目
- 好試解籤** 重點歸納、時事修法以及命題趨勢提醒。
- 達人推薦** 張逸仙 普考地政
高點總複習課程不僅可以快速複習重點，命中率也很高！我特別推薦許文昌跟于俊明老師，教學認真、教材豐富，非本科系的考生也能快速上手，讀書更有效率！



三等 **5,000** 元 定價 8,000元起
四等 **4,000** 元起

大吉

題庫班 → 打造高分力

- 求勝科目** 經濟學/財政學/稅法/會計/審計/政會
- 好試解籤** 名師嚴選經典考題，傳授看題能力以及教導高分答題技巧！
- 達人推薦** 柯辰穎
高普考財稅行政雙榜
隨著考期越來越近，我開始感到心慌，所以跑去報名會計&經濟&財政的題庫班，老師解題讓我釐清觀念，增加解題能力。



1,800 元起/科
4堂/科 定價 5,000元

高點 · 高上

高普考 衝刺

商資 · 地政 / 必勝錦囊

考運亨通

大吉

申論寫作班 → 論正寫題力

- 求勝科目** 審計/民法
- 好試解籤** 課前練題，高質量批改服務，建立答題架構，提高寫作高分力！
- 達人推薦** 李濤亦 高普考會計雙榜
高點老師請申論題命中率非常高！審計公報後期時間不太凶，只抓老師重點來背，申論竟拿到**32分**！



2,500 元/科
6堂起/科 定價 5,000元

大吉

公經進階班 → 鞏固強試力

- 好試解籤** 透析考題趨勢，加強進階內容，使考生能進一步掌握艱深考題。
- 達人推薦** 陳樂庭 高普考經建行政【狀元】
推薦張政(張家璋)老師的公經進階課程，他用數理詳細說明觀念，讓我實力大增！



2,500 元

以上考場優惠 110/10/20 前有效，限面授/VOD，當期最新優惠洽各分班櫃檯或高上生活圈！



另有**行動版課程**隨時可上
試聽&購課，請至

1 知識達購課館
ec.ibrain.com.tw



2 高點網路書店
publish.get.com.tw



《程式語言》

一、語意分析 (Semantics Analysis) 對於程式語言的正確執行非常重要。

(一) 以下是Java和C++程式，請說明程式編譯、執行結果與其原因，以及程式指令之意義或影響。(18分)

(1) Java 程式

```
static void test1(){
    int n;
    int [] x = new int[n];
}
```

(2) Java 程式

```
static void test2(){
    int n=0;
    int [] x = new int[n];
}
```

(3) Java 程式

```
static void test3(){
    int n=0;
    int [] x = new int[n];
    x[0]=0;
}
```

(4) C++ 程式

```
void test4(){
    int n;
    int x[n];
}
```

(5) C++ 程式

```
void test5(){
    int n=0;
    int x[n];
    x[0] =0;
}
```

(6) C++ 程式

```
void test6(){
    int n=0;
    int *x = new int[n];
    x[0] =0;
}
```

【版權所有，重製必究！】

(二) 請說明Java與C++語言在陣列宣告上的語意分析的方法，與其優缺點。(7分)

試題評析

語意分析(Semantics Analysis)在程式語言當中是非常重要的，本題使用Java與C++兩個語言來作為比較，以陣列來說，Java是比較純的物件導向語言，將陣列以物件方式處理，所以產生物件後，有關陣列的相關資訊一併產生，如陣列長度等，如果使用陣列不當，如超出索引範圍，則會發生Array Index Out Of Bound Exception異常。反觀C++有兩種產生陣列用法，一種使用陣列指令，如

| | |
|------|---|
| | int a[10];，另一種使用動態配置法，可以需要陣列時當時再產生，如int *p=new int[10];，如超出索引使用範圍，則看是不是侵犯記憶體使用，如果是，則發生Segmentation Fault。本題熟悉Java與C++語言的考生容易獲得高分。 |
| 考點命中 | 1.《高點·高上程式語言講義》第一回，許振明編撰，p.78~p.79 2.《高點·高上程式語言講義》第二回，許振明編撰，p.7 3.《高點·高上程式語言講義》第五回，許振明編撰，p.35~p.39 |

答：

(一)

(1)Java 程式

```
static void test1(){
    int n;                // n值沒有預設值
    int [] x = new int[n]; // 陣列無法使用new配置記憶體空間，程式編譯錯誤。
}
```

(2)Java 程式

```
static void test2(){
    int n=0;              // n值預設值為0
    int [] x = new int[n]; // 陣列動態配置整數陣列但空間為0，程式可編譯可執行。
}
```

(3)Java 程式

```
static void test3(){
    int n=0;              // n值預設值為0
    int [] x = new int[n]; // 陣列動態配置整數陣列但空間為0，程式可編譯可執行。
    x[0]=0;              // 執行時出現陣列超出範圍 ArrayIndexOutOfBoundsException。
}
```

(4)C++程式

```
void test4(){
    int n;                // n值沒有設值，預設值為0。
    int x[n];             // 陣列長度為0，沒有空間可以儲存整數，可編譯可執行。
}
```

(5)C++程式

```
void test5(){
    int n=0;              // n值沒有設值，預設值為0。
    int x[n];             // 陣列長度為0，沒有空間可以儲存整數，可編譯可執行。
    x[0] =0;             // x[0]的值設為0，如果列印x[0]會顯示0。
}
```

(6)C++程式

```
void test6(){
    int n=0;              // n值沒有設值，預設值為0。
    int *x = new int[n]; // 整數指標x動態配置給陣列的記憶體空間，可編譯可執行。
    x[0] =0;             // x[0]的值設為0，如果列印x[0]會顯示0。
}
```

【版權所有，重製必究！】

(二)

- (1)Java的陣列是屬於物件型別，使用new指令配置空間給陣列並使用參考變數控制，陣列長度經宣告後就不能改變。使用物件方式配置陣列的好處是可以由陣列物件得到陣列相關的資訊，如陣列長度。
- (2)C++可以使用指令配置陣列，如 int x[10]，也可使用動態配置記憶體給陣列，如 new int[10]。指令配置需確認陣列長度後就不能改變，動態配置則是在執行時要使用陣列才執行new指令配置指定的記憶體空

間，所以動態配置比較具有彈性。

二、程式驗證的應用。

- (一)請說明「測試驅動發展方法 (Test Driven Development, TDD)」的概念及優點。(7分)
 (二)有一MySort類別的方法int[] binarySort(int data[])，將陣列data內的資料由小到大排序後回傳，請依據TDD的概念設計測試案例。(10分)
 (三)請以Java/JUnit語言完成以下測試程式 (I) ~ (II)。(8分)

```
@Test
public void testBinarySort(MySort (I) ){
    int[] source = {2, 3, 5, 9, 12, 7};
    int[] target = obj.binarySort(source);
    for(int i=0; i<source.length-1; i++){
        assertTrue(target[i] < (II) );
    }
}
```

| | |
|------|---|
| 試題評析 | 測試驅動開發TDD(Test-Driven Development)的精神是「先寫測試再開發」，主要是「模擬外部如何使用目標程式，驗證目標程式的行為是否符合預期」。TDD可以幫助程式在開發初期能幫助介面設計，做好user friendly使用者友善。因此，許多專案會導入TDD作為開發軟體程式的機制，考生如果參與過TDD的專案則容易獲得高分。 |
| 考點命中 | 1.《高點·高上程式語言講義》第一回，許振明編撰，p.83~p.88 2.《高點·高上程式語言講義》第五回，許振明編撰，p.78 |

答：

(一)測試驅動開發TDD(Test-Driven Development)是一種開發流程，先寫測試再開發，先寫測試除了能確保測試程式的撰寫，另一個好處是有助於在開發初期釐清程式介面如何設計。測試程式的作用是「模擬外部如何使用目標程式，驗證目標程式的行為是否符合預期」。

TDD 流程五步驟:

- 步驟一：選定一個功能，新增測試案例
 步驟二：執行測試，得到 Failed(失敗)
 步驟三：實作「夠用」的產品程式
 步驟四：再次執行測試，得到 Passed(成功)
 步驟五：重構程式

(二)

(1)確認使用者需求，撰寫測試案例

函數：testBinarySort傳入一個MySort類別物件，透過物件呼叫binarySort()方法。此方法可以將陣列source排序後回傳給陣列target。

(2)撰寫測試案例程式: testBinarySort，用此方法驗證傳入的資料是否能一小到大排序後回傳。

(三)

(I) obj

(II) target[i+1]

三、建構股票交易資料庫 (Stock)，請寫出SQL 指令。

客戶表格 (Customer)。

| | | | |
|---------------------------------|------------------------------|---------------------------------|--------------------------------|
| 客戶編號 (cid) [整數、主鍵] [自動增加] | 客戶姓名 (cname) [少於10字元可變字串] | 客戶帳戶餘額 (balance) [整數、非空值] | 客戶融資餘額 (margin) [整數、非空值] |
|---------------------------------|------------------------------|---------------------------------|--------------------------------|

證券交易表格 (StockTrade)

| | | | | |
|-----------------------------------|---------------------------|---------------------------------|-------------------------------|------------------------|
| 交易編號 (id) [整數、主鍵] [自動增加] | 證券編號 (sid) [整數、非空值] | 證券每股購入價格 (price) [整數、非空值] | 證券購入股數 (share) [整數、非空值] | 客戶編號 (cid) [整數、非空值] |
|-----------------------------------|---------------------------|---------------------------------|-------------------------------|------------------------|

(一) 造出Customer, StockTrade表格。(10分)

```
CREATE TABLE Customer ( _____ );
CREATE TABLE StockTrade ( _____ );
```

(二) 查詢客戶姓名是"Tom"所有購買股票編號與購入總股數。(5分)

(三) 撰寫Store Procedure，造出一個暫時的資料表Report，含兩個整數資料欄位（證券編號sid, 證券價格price）；加入10筆資料，再根據證券價格由小到大排序，查詢列出此10筆資料。(10分)

```
delimiter $$
CREATE PROCEDURE x()
BEGIN
    DECLARE i INT DEFAULT 1;
    _____
END$$
```

| | |
|------|---|
| 試題評析 | 資料庫語言SQL是資訊人必學的語言，無論你是DBA或軟體工程師。其中預存程式(Stored Procedure)是利用程式來提供控制資料庫的重要用法，幾乎每個資料庫都會使用的功能。本題主要測試建立資料表、查詢資料表與撰寫Stored Procedure，這些都是基礎與必需了解的主題，本題考生只要小心作答就可獲取高分。 |
| 考點命中 | 《高點·高上程式語言講義》第五回，許振明編撰，p.108~p.115 |

答：

(一)

```
CREATE TABLE Customer(
    cid INT NOT NULL AUTO_INCREMENT,
    cname VARCHAR(10),
    balance INT NOT NULL,
    margin INT NOT NULL,
    PRIMARY KEY(cid)
```

);

```
CREATE TABLE StockTrade(
    id INT NOT NULL AUTO_INCREMENT,
    sid INT NOT NULL,
    price INT NOT NULL,
    share INT NOT NULL,
    cid INT NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY(cid) REFERENCES Customer(cid)
```

);

(二)

```
SELECT Customer.cname, StockTrade.sid, SUM(StockTrade.share) AS Total
FROM Customer INNER JOIN StockTrade ON Customer.cid=StockTrade.cid
```

【版權所有，重製必究！】

```
WHERE Customer.cname = 'tom'
GROUP BY Customer.cname, StockTrade.sid;
```

(三)

```
CREATE TABLE Report(sid INT NOT NULL AUTO_INCREMENT, price INT NOT NULL, PRIMARY KEY(sid));
WHILE i <= 10 DO
    INSERT INTO Report values(i, i*10);
    SET i=i+1;
END WHILE;
SELECT * FROM Report ORDER BY price ASC;
```

四、程式例外處理的設計對於資訊系統的可靠性非常重要。

(一)請完成以下C++程式 (I) ~ (V) 指令，處理兩數相除的例外狀況，使輸出為：(15分)

```
Exception:empty
Exception:not a number
Quotient:Exception:divided by zero
Quotient:2.4
```

```
#include <iostream>
#include <exception>
#include <string.h>
#define N 10
using namespace std;
class EmptyException:public exception {
public:
    virtual const char* what()const throw(){
        _____ (I) _____;
    }
};
class NotNumberException:public exception {
public:
    virtual const char* what()const throw(){
        _____ (II) _____;
    }
};
class DividedByZeroException:public exception {
public:
    virtual const char* what()const throw(){
        _____ (III) _____;
    }
};
int valid(const char x[N]){
    int result=0;
    if(strlen(x)==0)throw EmptyException();
    for(int i=0; i<strlen(x); i++){
```

【版權所有，重製必究！】

```

        if(!isdigit(x[i]))
            throw NotNumberException();
        result = _____ (IV) _____;
    }
    return result;
} double quotient(int n1, int n2){
    if(____ (V) _____)
        throw DividedByZeroException();
    return static_cast<double>(n1/n2);
}
void test(const char x1[N], const char x2[N]){
    int n1, n2;
    try {
        n1=valid(x1);
        n2=valid(x2);
        cout<<"Quotient:"<<quotient(n1, n2);
    }
    catch(EmptyException &e){
        cout<<"Exception:"<< e.what();
    }
    catch(NotNumberException &e){
        cout<<"Exception:"<< e.what();
    }
    catch(DividedByZeroException &e){
        cout<<"Exception:"<< e.what();
    }
    cout<<endl;
}
int main(){
    test("", "");
    test("a", "12");
    test("10", "0");
    test("12", "5");
    return 0;
}

```

(二)請說明使用try-catch與if-else，處理例外狀況的優缺點。(5分)

(三)請說明C++與Java在try-catch中finally設計的異同與其理由。(5分)

| | |
|-------------|---|
| 試題評析 | 異常處理(Exception)在物件導向(Object-Oriented)中是一個重點主題，主要的優點在於許多的異常狀況無論在編譯時期(Compile-time)或執行時期(Run-time)都有可能發生，常常發生後程式就結束，無法繼續執行下去，異常處理就是解決這個問題的最佳機制，只要規劃好可能發的狀況，事先提供解決方案，就可以引導程式排除異常，讓程式可以繼續執行，不需要離開程式，本題考生只要小心作答就可獲取高分。 |
| 考點命中 | 1.《高點·高上程式語言講義》第一回，許振明編撰，p.109~p.112 2.《高點·高上程式語言講義》第二回，許振明編撰，p.45~p.47 3.《高點·高上程式語言講義》第五回，許振明編撰，p.73~p.77 |

答：

(一)

- (I) return "empty";
- (II) return "not a number";
- (III) return "divided by zero";
- (IV) $(x[i]-'0')+(result*10)$;
- (V) $n2==0$

(二)if..else是程式指令，由條件判斷後決定後，條件True執行if後的區塊敘述，條件False執行else後的區塊敘述。

try..catch主要用在異常(Exception)處理，需要經過測試的程式碼放在try區塊中，同時準備程式可能發生的異常狀況，將各種解決方案寫在各個catch中。如果有必定執行的指令則放在finally區塊內。

當發生異常(Exception)時，就會按照預先寫好的catch規則處理，處理完成後，程式會繼續執行後續的程式碼，不會跳離整個程式。

(三)C++與Java均支援異常(Exception)處理機制。

相同點：

- (1)語法都是try..catch
- (2)處理完成後，程式皆可繼續執行下去。

相異點：

- (1)Java有支援finally功能，但C++無。
- (2)Java有規定必要處理的Exception，如IOException，也就是checked Exception，C++並無此規定。

【版權所有，重製必究！】