

類 科：資訊處理
科 目：程式語言
考試時間：2小時

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)本科目除專門名詞或數理公式外，應使用本國文字作答。

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;  
}
```

(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)

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)___);
    }
}
```

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

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++){
        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)