

程式設計（107-1）

作業十

作業設計：楊其恆、孔令傑
國立臺灣大學資訊管理學系

繳交作業時，請至 PDOGS (<http://pdogs.ntu.im/judge/>) 為第一題做同儕互評，再為第二至四題各上傳一份 C++ 原始碼（以複製貼上原始碼的方式上傳）。每位學生都要上傳自己寫的解答。不接受紙本繳交；不接受遲交。請以英文或中文作答。

這份作業的截止時間是 **2017 年 12 月 22 日凌晨一點**。在你開始前，請閱讀課本的第 11 章¹。為這份作業設計測試資料並且提供解答的助教是楊其恆。

第一題

(0 分) 請在 PDOGS 上批改你被隨機分配到的作業九第四題的程式碼，根據它在正確性以外的部份給它 1 至 5 分的評分，並且說明你給分的依據。建議在評分時參考以下六個面向。在前五個面向上，一個面向上做得好就得一分，還不錯則半分，不好則零分；在第六個面向上則在有必要時扣分。六個面向的分數合計後無條件進入即為你最後給的總分。

- 可讀性：變數與函數名稱是否具有合適的資訊量？程式碼排版是否良好且具有前後一致性？是否有合適的註解？關於註解，當然不需要每一行都有註解，但若你發現在某一大段落裡都沒有註解，或某個你感覺很不易看懂的部份沒有註解，你可以指出來；不要直接說「註解太少」但沒有說是哪邊缺乏註解。
- 模組化程度：是否有宣告合適的函數？是否有避免將非常類似的程式片段寫複數次而非寫成函數？是否有避免一個函數做非常多事情？函數間是否有合適的 decoupling？直接閱讀 main function 是否能很快地理解程式在大方向上的運算邏輯？
- 效率：程式運算是否有合理的運算效率？當然我們不要求每個同學都寫出超級有效率的精妙演算法，但至少一個程式不應該進行過多不必要的運算，也不應該耗用過多不必要的記憶體空間。如果你看不出這個程式的效率有明顯的問題，我們建議你直接給一分。
- 擴充性：當要解的問題變得更複雜的時候，我們能不能簡單地修改這個程式以解決新的問題，而不是寧可砍掉重練？這個議題當然也很主觀，所以如果你不能明確地指出在怎樣的新問題上，這個程式會有擴充性問題，我們建議你直接給一分；如果你不能指出很嚴重的問題，我們建議你至少給半分。但對批改者來說，這個關於擴充性的思考其實是很好的訓練。試試看吧！
- 其他：如果有任何其他令你想扣分的理由，請明確地寫出來並且在這個面向上扣分；沒有的話就給一分。
- 題目規範：你應該檢查那份程式碼有沒有違反題目的規範，如果有（例如題目說不可以用上課沒教過的東西，但他用了，或者題目說一定要用指標和動態記憶體配置，但他沒用），就給他零分。當然，請明確地指出他哪邊違反了題目的規範。

¹課本是 Deitel and Deitel 著的 *C++ How to Program: Late Objects Version* 第七版。

該題其中 10 分取決於檢視你的程式碼的同學給你的分數總和（必要時助教會出來主持公道，請不用緊張），另外 10 分取決於你對同學的程式碼的評語和評分的合理性和建設性（原則上除非被申訴，且助教檢視後發現你確實評得很不公平，否則只要有評就會得到 10 分）。

第二至四題的前言

這次作業與上次類似，只是在本次作業，我們除了考慮不同種類的資料型態外，還要加入 WHERE 子句的處理。大家所拿到的作業會包含以下幾個檔案：

1. 你正在閱讀的作業說明檔案。
2. 「sampleCode2.cpp」：助教寫好的程式骨架，請大家從此份程式開始延伸。程式中會標記要完成本次作業，所需要完成的部分。對於函數的實作方法，也可參考其中的註解。這份檔案會在作業九截止後公布新的版本，包含了上次作業中大家應該完成的部分之程式碼（檔案中被標記 you can use your own code 之處），如果上次作業進行得不是很順利，可以直接由這份開始這次作業。請注意這份檔案中的 `strucure`、`loadcsv` 與 `print` 函數實作方法加入了本次作業所需的內容，因此若大家要直接使用上次作業中自己寫的程式來增加功能，請記得更新這些定義。
3. 「table1.csv」、「table2.csv」：原始資料會存在一些 CSV（comma-separated values）檔案中，在 PDOGS 上供你的程式存取並且做測試。在你自行開發程式的期間，你也需要一些測試用的資料檔，這就是這兩個 CSV 檔的用途了。程式中的 `loadcsv` 函數可以將這些資料讀入。

與上次作業相同，我們將對題目中會出現的 MySQL query 做以下的限制或簡化：

1. 最多只會出現 SELECT、FROM、WHERE 三種子句，且必定照順序出現。
2. SELECT 子句後只會出現欄位名稱，不包含 SQL function 如 COUNT、SUM 等，但可能出現「*」（表示所有欄位）。
3. 資料型態包含整數與字串二種。
4. 在指令中表示字串時，不會使用引號「」。
5. 資料中不會出現空值。
6. 查詢指令必定會符合 MySQL 的文法。

在本次作業中，我們要處理 SELECT、FROM 及 WHERE 子句，並且考慮整數與字串型態的資料內容。

第二題

（20 分）在本題中，我們先嘗試把資料型態擴展為整數與字串二種。為了達成這個目的，需要對程式碼做了一些修改，首先增加一個 `structure Data`，取代上次作業中大家用來儲存字串所使用的 `char*` 型態²

²由於對於同一欄位中的資料，其型態只能是整數或字串其一，因此這個 `structure` 中的二個變數並不會同時被使用到，要達成這個目的，在 C++ 中使用 `union` 是一個更好的方法，有興趣可以自行查詢其意義。

```

struct Data
{
    int i;
    char* c;
};

```

另外將 struct Field 中代表資料行的變數型態由 char** 改為 Data*。並新增一個變數 isString 表示該欄位的資料型態是不是字串。

```

struct Field
{
    bool isString;
    char* name;
    Data* data;
};

```

接著為了使助教的 loadcsv 函數可以在讀入資料時正確地判斷資料型態，大家需要再完成一個函數，loadcsv 中會直接呼叫這個函數：

```

bool isStringType(char* str);

```

傳入一個字串，這個函數要判斷他是不是一個整數，如果是回傳 false，不是則回傳 true。在本題的測試資料中，所有的字串都會包含至少一個數字字元與減號「-」的字元，亦即只要你看到任何一個字元不是數字或減號，它就是字串；所有字元都是數字或減號且最多只有第一個字元是減號時，它就是整數。請注意在 print 函數中，整數或字串的輸出格式是不同的，因此如果你的型態判斷錯誤，會造成輸出結果不同，也就會發生 Wrong Answer。

輸入輸出格式與上次作業中的第三題相同。系統會提供一共 10 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中，會有一個字串，代表輸入資料（一張新的資料表）的檔名。讀入這些之後，請使用 loadcsv 函數讀入資料，並使用 print 函數將資料印出。

舉例來說，如果輸入是

```

table1.csv

```

則輸出應該是

```

-----
| field1 | field2 | field3 | field4 |
-----
| aaa    | xxx    | -15    | 37g9    |
| bbb    | yyy    | 27     | 1r      |
| ccc    | zzz    | -39    | a4w     |
-----
3 rows in set

```

第三題

(20 分) 可以使用二種資料型態讀入並輸出表格資料後，接著我們要讓它一樣可以接受包含 `SELECT` 與 `FROM` 子句的查詢指令並印出結果。要達到這個目的，請大家修改上次作業中所寫的函數，使之可以處理二種資料型態的查詢。

系統會提供一共 10 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中，會有 $n + 2$ 行，第一行包含一個整數 n ，代表總共有幾張表。第二行開始的 n 行之中，每一行包含二個字串。第 $i + 1$ 行的第一個字串代表第 i 張表的名稱，第二個字串代表該表所在的檔案名稱，請使用 `loadcsv` 函數讀取這個檔案中的資料。二個字串以一個空白字元分隔。第 $n + 2$ 行包含一個字串，代表要被執行的指令。其中 $1 \leq n \leq 5$ ，名稱字串的長度不超過 10 個字元，檔名字串不超過 100 個字元，且二者不含有空白字元。讀入資料後，請根據指令內容，輸出應有的查詢結果。

舉例來說，如果輸入是

```
2
table1 D:/table1.csv
table2 D:/table2.csv
SELECT firstName, lastName, age FROM table2;
```

則輸出應該是

```
-----
| firstName | lastName | age |
-----
| Lois      | Walker   | 36  |
| Theresa   | Lee      | 25  |
| Ruby      | Rogers   | 37  |
| Carolyn   | Hayes    | 59  |
| Anne      | Rogers   | 28  |
| Daniel    | Cooper   | 28  |
| Donna     | Brown    | 34  |
| Carl      | Collins  | 44  |
| Alan      | Rogers   | 39  |
-----
9 rows in set
```

這一題的 20 分會根據程式運算的正確性給分。PDOGS 會編譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。一筆測試資料佔 2 分。

第四題

(60 分) 最後我們在查詢指令中加入 `WHERE` 子句。在範例程式中，我們增加了一個 `structure Condition`，表示 `WHERE` 子句所代表的意義。

```

struct Condition
{
    char* fieldName;
    Data value;
};

```

另外在 structure `Command` 之中，也增加了一個變數 `condition`，代表 `WHERE` 子句的條件，大家可以假設 `parseCommand` 函數會正確地將查詢指令中的 `WHERE` 子句之意義存到該變數中。若查詢中不包含 `WHERE` 子句，則 `condition` 中的 `fieldName` 會被設為 `nullptr`。

```

struct Command
{
    int selectNum;
    char** select;      //select
    char* tableName;    //from
    Condition condition; //where
};

```

請大家修改 class `Table` 中的成員函數 `executeCmd`，使之可以處理包含 `WHERE` 子句的查詢指令。

系統會提供一共 20 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中，會有 $n + 2$ 行，第一行包含一個整數 n ，代表總共有幾張表。第二行開始的 n 行之中，每一行包含二個字串。第 $i + 1$ 行的第一個字串代表第 i 張表的名稱，第二個字串代表該表所在的檔案名稱，請使用 `loadcsv` 函數讀取這個檔案中的資料。二個字串以一個空白字元分隔。第 $n + 2$ 行包含一個字串，代表要被執行的指令。其中 $1 \leq n \leq 5$ ，名稱字串的長度不超過 10 個字元，檔名字串不超過 100 個字元，且二者不含有空白字元。讀入資料後，請根據指令內容，輸出應有的查詢結果。

舉例來說，如果輸入是

```

2
table1 D:/table1.csv
table2 D:/table2.csv
SELECT firstName, lastName, age FROM table2 WHERE lastName = Rogers;

```

則輸出應該是

```

-----
| firstName | lastName | age |
-----
| Ruby      | Rogers   | 37  |
| Anne      | Rogers   | 28  |
| Alan      | Rogers   | 39  |
-----
3 rows in set

```

如果輸入是

```
2
table1 D:/table1.csv
table2 D:/table2.csv
SELECT ID, lastName, age, City FROM table2 WHERE age = 28;
```

則輸出應該是

```
-----
| ID      | lastName  | age | City      |
-----
| a253    | Rogers    | 28  | Stockholm |
| a480    | Cooper    | 28  | Manning   |
-----
2 rows in set
```

如果輸入是

```
2
table1 D:/table1.csv
table2 D:/table2.csv
SELECT * FROM table2 WHERE State = MD;
```

則輸出應該是

```
Empty set
```

你上傳的原始碼裡應該包含什麼

你的.cpp 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你**不可以**使用上課沒有教過的方法，但上課提過的 library 裡面的所有功能都可以用。除此之外，你**一定要**用 class 以及完成題目指定的任務。

評分原則

- 這一題的其中 40 分會根據程式運算的正確性給分。PDOGS 會編譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。一筆測試資料佔 2 分。
- 這一題的其中 20 分會在作業十一中被評定。屆時我們會讓同學們互相檢視彼此的本題程式碼，並且就可讀性、易維護性、模組化程度、排版等面向寫評語和給評分（當然一切都是匿名的）。該任務在本題中會佔 20 分，其中 10 分取決於檢視你的程式碼的同學給你的分數（必要時助教會出來主持公道，請不用緊張），另外 10 分取決於你對同學的程式碼的評語和評分的合理性和建設性。若你在本次作業中完全沒有寫這一題，那屆時自然沒有人能檢視你的程式碼，你也就得要損失這 10 分了。

此外，由於本題是要求使用 class，所以你必須要寫 class 並且實做題目中要求的那兩個函數，然後適當地使用它們。如果屆時你負責評分的程式沒有做到這件事，請給他 0 分；如果你的程式沒做到這件事，你也就會損失本題這 10 分了。

第五題 (bonus)

(20 分) 針對以下十題是非題，我們會使用 PDOGS 自動批改，因此請寫一個 C++ 程式，內容就是先讀入一個整數，若讀入的數字為 i ，則印出第 i 小題的答案，若為是則印出 1、若為否則印出 0。舉例來說，如果題目只有四題，且你認為答案依序是是、否、是、是，則你上傳的程式碼應該是

```
#include<iostream>
using namespace std;

int main()
{
    int problem = 0;
    cin >> problem;
    if(problem == 1)
        cout << 1;
    else if(problem == 2)
        cout << 0;
    else if(problem == 3)
        cout << 1;
    else
        cout << 1;

    return 0;
}
```

PDOGS 會餵給你的程式的，一定是 1、2 直到 10 這十個整數。有別於作業中一般的程式題，本題在你上傳程式碼時，測試資料是還沒有放上 PDOGS 的，助教會等作業截止後才上傳測試資料（和答案）到 PDOGS 並重新批改此題。換言之，你上傳程式碼時是不會顯示你得幾分的，更不會顯示你對或錯哪些筆測試資料。你會看到你得 0 分，但此數字在助教重新批改之後就會被更新成正確的分數了。

以下題目如果沒有特別指名，請用 C++ 為基準作答。若你看到一段程式碼，請假設他們是被寫在一個有良好且必備的 include 敘述、using namespace 敘述的程式的結構正確的 main function 裡面。

- (a) Constant object 可以呼叫 constant instance function，不能呼叫 non-constant instance function。
- (b) Non-constant object 可以呼叫 non-constant instance function，不能呼叫 constant instance function。
- (c) 在一個 class 的 instance function 中，**this** 是 calling object 的位址，***this** 是 calling object 的本體。

- (d) 在一個 class 的 static function 中，`this` 是 calling object 的位址，`*this` 是 calling object 的本體。
- (e) 一個函數如果回傳一個 constant object，則此回傳的 object 不能在 caller 裡面被放在 assignment operator 的左邊。
- (f) 在 C++ 裡面做 operator overloading 時，不能更改 operator 的 associativity 與運算元個數。
- (g) 如果針對一個 class 你有理由自行定義 copy constructor 和 destructor，你就有理由自行定義 assignment operator。
- (h) `ifstream` 和 `ofstream` 是兩個 C++ 標準函式庫裡面的物件。
- (i) 如果程式設計師自行定義了 copy constructor，這個 class 就不會有系統預設的 default constructor。
- (j) 如果程式設計師自行定義了 default constructor，這個 class 就不會有系統預設的 copy constructor。

小提醒：在 PDOGS 上面讓大家繳交此題的地方，會有兩組「與上面正式要計分的題目完全無關的」範例輸入輸出，純粹是用來讓大家確認自己那個被批改的 `if-else` 程式是可以被正確執行的。請確認你的程式在針對範例輸入輸出做撰寫後，能讓你在這一題得到「Accepted」，接著再去針對要計分的題目把你的正確答案寫上去然後繳交。當然，即使你曾經看到「Accepted」，也不代表你繳交的題目在這題已經得到滿分了。