

程式設計 (105-2)

作業九

作業設計：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、二題各上傳一份 C++ 原始碼 (以複製貼上原始碼的方式上傳)、對第三題做同儕互評，再為第四題上傳一個 PDF 檔。第四題是 bonus 加分題。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 5 月 15 日凌晨一點**。在你開始前，請閱讀課本的第 9-11 章¹。為這份作業設計測試資料並且提供解答的助教是李昱賢 (Rick Lee)。

第四題 (bonus)

(20 分，各 5 分) 請根據第一題和第二題的題目回答下列問題：

- (a) 我們是否應該為 RationalNumber 實作 copy constructor 或 destructor，或者 overload assignment operator？為什麼？

Ans: 因為在 RationalNumber 這個 Class 中，我們的 data 都是 store by value，因此可以不用特別去實作 copy constructor 及 destructor

- (b) Student 滿足一些讓我們為其實作 copy constructor、destructor 和 assignment operator 的理由，但實際應用上我們既不會複製一個學生，也不會把一個學生 assign 成另一個學生。如果我想要避免 default assignment operator 被呼叫，但又不想實作一個 assignment operator (因為根本不應該提供這個功能)，我應該怎麼做？請把程式碼寫下來並且說明原因。

Ans: 我們可以實際定義 assignment operator，這樣就不會自動產生 default 的 copy constructor，而且我們將這些 operator 定義在 private 內，使用時也沒辦法 call 到

```
class Student
{
private:
    string studentID;
    int courseCnt;
    CourseGrade* gradeList;
    Student(const Student&);
    Student& operator=(const Student&);
};
```

- (c) 承上題，請實作 Student 的 destructor。

Ans:

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

```
Student::~~Student(){
    delete [] gradeList
}
```

- (d) 承上題，事實上我們是需要自己實作 copy constructor 的，因為「應該用得到」。請說明在學校裡事實上沒有複製人的情況下，Student 的 copy constructor 何時會被用到？接著請實作這個 copy constructor 去實現 deep copy。

Ans: 在學校時可能會遇到學生轉系的狀況，在每個系上會保留系上學生的修課紀錄，因此當學生轉系時，新轉過去的系，會需要將學生原本的資料 copy 過去，因此會需要用 copy constructor

```
Student::Student(const Student& s){
    this->studentID = s.studentID;
    this->courseCnt = s.courseCnt;
    this->gradeList = new CourseGrade[this->courseCnt];
    for (int i = 0; i < this->courseCnt; i++)
        this->gradeList[i] = s.gradeList[i];
}
```