

Programming Design, Spring 2015

Homework 12B

Instructor: Ling-Chieh Kung
Department of Information Management
National Taiwan University

To submit your work, please upload two CPP files for Problems 1 and 2 (optional) to PDOGS at <http://pdogs.ntu.im/judge/>. Each student must submit her/his individual work. No hard copy. No late submission. The due time of this homework is 8:00am, June 15, 2014. Please answer in either English or Chinese.

This is Homework 12B; Homework 12A has been given in the next week. They together count for 100 basic points and 40 bonus points. Please note that they have separated submission deadlines.

Before you start, please read Chapters 13 of the textbook.¹ The TA who will prepare the solution for this homework is Willy Liao.

Problem 1

(50 points) In this problem, let's practice on polymorphism by repeating what we did in Problem 1 of Homework 12. In particular, note that in the suggested solution, we create two arrays for the two types of transactions:

```
//store all the transactions in two ways
FirstTSN** ft = new FirstTSN*[1000 * fileNum];
int ftInd = 0;
SecondTSN** st = new SecondTSN*[1000 * fileNum];
int stInd = 0;
```

where `fileNum` is the number of data files for one program execution. Please note that 1000 times `fileNum` is the maximum number of *total* transactions (including both types); we will definitely waste half of the spaces we prepare for transactions! Moreover, when you want to access a transaction, you will need to know the type of this transaction for you to identify one array to search in. That complicates your program (especially if you have many types of transactions).

Fortunately, now you know polymorphism helps. You may simply create one array

```
AbstractTSN** at = new AbstractTSN*[1000 * fileNum];
```

to store both types of transactions. If you want to invoke a transaction's member function, you may even utilize function overriding, virtual functions, and late binding to access the right version of that member function. Polymorphism can help you significantly simplify your program (if you use it in the right way).

Input/output formats

The input and output requirements are exactly the same as Problem 1 in Homework 12A.

What should be in your source file

Your program still needs to have the three classes defined in Problem 1 of Homework 12A. Moreover, you need to modify your program used in Homework 12A to utilize polymorphism.

¹The textbook is *C++ How to Program: Late Objects Version* by Deitel and Deitel, seventh edition.

Your .cpp source file should contain C++ codes that will both read testing data and complete the above task. For this problem, you are NOT allowed to use techniques not covered in lectures. You should write relevant comments for your codes.

Grading criteria

30 points for this program will be based on the correctness of your output. PDOGS will compile your program, feed testing data into your program, and check the correctness of your outputs. For each set of input data, if your program outputs correctly without violating the space limit, you get 2 points.

20 points for this program will be based on how you write your program, including the logic and format. In particular, we will consider how you utilize polymorphism. Please try to write a robust, efficient, and easy-to-read program.

Problem 2 (bonus)

(20 points) Continue from Problem 1. Redo the task given in Problem 2 of Homework 12A. There will be 10 input files, one for 2 points. For this problem, you may use any technique you like.