TCP1201 Objected-Oriented Programming and Data Structures Assignment 2

Trimester 1, Session 2015/2016
Faculty of Computing and Informatics
Multimedia University

DUE DATE: 21Sept 2015 (Monday), 11:59pm

Outline

This assignment contributes 10% of the total course marks. This assignment consists of only one question. Every student is to submit one assignment individually.

You are strongly advised to submit the assignment on time even though it is incomplete. Late submissions will be given a zero mark. All assignments (across both TC01 and TC02) will be subject to a plagiarism checker, and those found to have plagiarized from other students' works or other sources will also be given a zero mark.

No extensions of dateline will be entertained so that the lecturers have sufficient time to assess and release the mark before the final exam.

If necessary, selected students will be called for an oral interview by your respective lecturer. The date and time of interview will be scheduled and announced after the assignment deadline.

Make sure your program code can be compiled under g++4.7.1. You may download Code::Blocks 13.12 with bundled g++ 4.8.1 at http://sourceforge.net/projects/codeblocks/files/Binaries/13.12/Windows/codeblocks-13.12mingw-setup.exe

Problem Statement

You have been assigned to set up a simple system for a bookstore to analyze and track the sales of books. In particular, a text file will be used to store data statistics of all the books sold. In order to achieve this, there are 2 main tasks in this assignment that you are to attempt.

1) SortedLList

Create a template based pointer-based linked list, **SortedLList** with the following functions:

- **SortedInsert** - Inserts a new node into the linked list such that the linked list is sorted in the ascending order based on the key.
- **Retrieve** - Retrieves a node that matches the target key value and return the information stored in the node.
- **Delete** - Deletes the node that matched the target key value.

For this task, write a simple demo program that is able to demonstrate these there new operations work as required. You may use simple data types such as integer or string.

Note:

- Please do proper separation of files (header file, implementation file and the main application file).
- You may add other basic and supporting (private) methods to **SortedLList** if needed.
- **Hint:** Since the nodes are inserted into the linked list in a sorted manner, the **Retrieve** and **Delete** functions should be efficient such that the function does not have to check the full linked list even if the target item does not exist.

2) **Book Sales System**

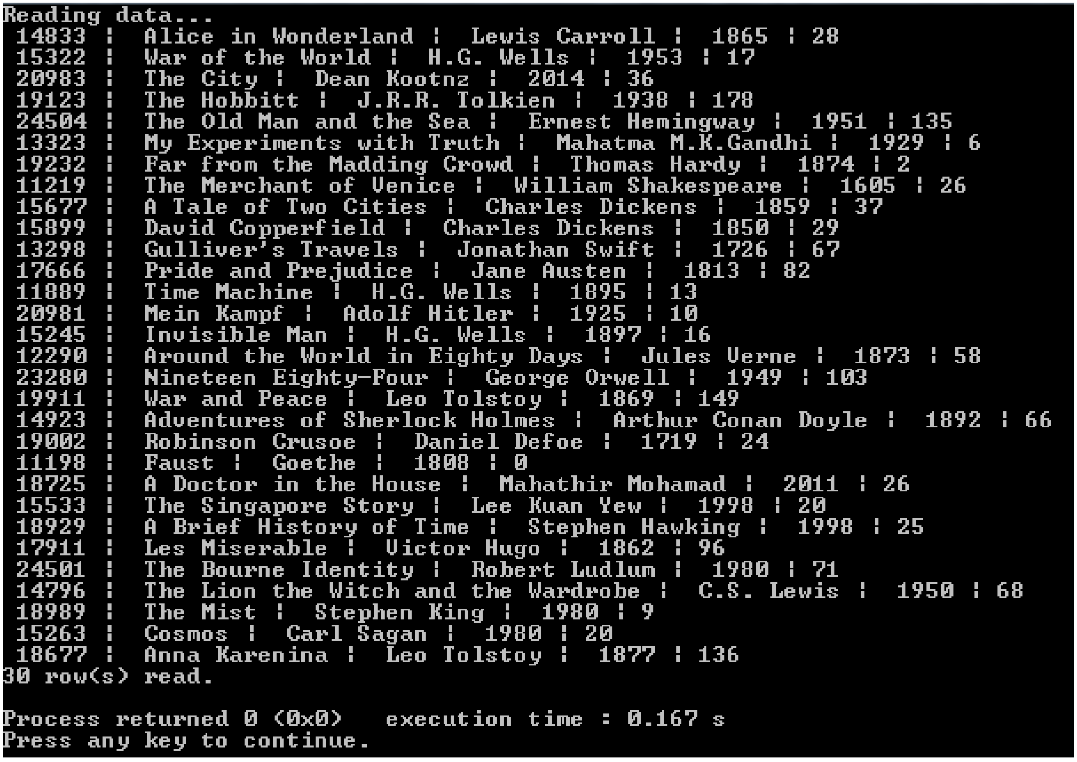
The book sales system should allow the bookstore owner to perform some useful sales analysis and statistics. In this task, you are to extract some important statistics from book data stored in a text file database. You are to use the provided class **BookItem** (**.hpp** and **.cpp** files), together with the **SortedLList** class that you have created in task 1, to create the following features and statistics:

- ○ **Book List** – List all books sorted by book title.
- ○ **Book Search** – Simple search feature that takes in a book title and returns other information of the searched book. You should return no matches, if the book does not exist.
- ○ **Top 10 best sellers** – Top 10 books that sold the most, in descending order of copies sold.

In the sample application file (***main.cpp***) provided, the STL vector is used to store the book details retrieved from the txt file (***db_small.txt***). In this assignment, you are required to replace the class with the **SortedLList** class that you have created in task 1.

readData() reads the database text file (***db_small.txt*** consisting of 20 entries only), displays rows of data that have been read, and then stores the data into the stack.

The output after compiling the sample application file is shown below:



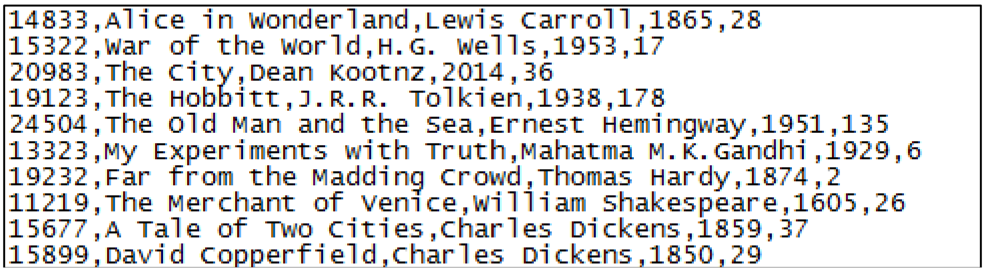
```
Reading data...
 14833 ¦  Alice in Wonderland ¦  Lewis Carroll ¦  1865 ¦ 28
 15322 ¦  War of the World ¦  H.G. Wells ¦  1953 ¦ 17
 20983 ¦  The City ¦  Dean Kootnz ¦  2014 ¦ 36
 19123 ¦  The Hobbitt ¦  J.R.R. Tolkien ¦  1938 ¦ 178
 24504 ¦  The Old Man and the Sea ¦  Ernest Hemingway ¦  1951 ¦ 135
 13323 ¦  My Experiments with Truth ¦  Mahatma M.K.Gandhi ¦  1929 ¦ 6
 19232 ¦  Far from the Madding Crowd ¦  Thomas Hardy ¦  1874 ¦ 2
 11219 ¦  The Merchant of Venice ¦  William Shakespeare ¦  1605 ¦ 26
 15677 ¦  A Tale of Two Cities ¦  Charles Dickens ¦  1859 ¦ 37
 15899 ¦  David Copperfield ¦  Charles Dickens ¦  1850 ¦ 29
 13298 ¦  Gulliver's Travels ¦  Jonathan Swift ¦  1726 ¦ 67
 17666 ¦  Pride and Prejudice ¦  Jane Austen ¦  1813 ¦ 82
 11889 ¦  Time Machine ¦  H.G. Wells ¦  1895 ¦ 13
 20981 ¦  Mein Kampf ¦  Adolf Hitler ¦  1925 ¦ 10
 15245 ¦  Invisible Man ¦  H.G. Wells ¦  1897 ¦ 16
 12290 ¦  Around the World in Eighty Days ¦  Jules Verne ¦  1873 ¦ 58
 23280 ¦  Nineteen Eighty-Four ¦  George Orwell ¦  1949 ¦ 103
 19911 ¦  War and Peace ¦  Leo Tolstoy ¦  1869 ¦ 149
 14923 ¦  Adventures of Sherlock Holmes ¦  Arthur Conan Doyle ¦  1892 ¦ 66
 19002 ¦  Robinson Crusoe ¦  Daniel Defoe ¦  1719 ¦ 24
 11198 ¦  Faust ¦  Goethe ¦  1808 ¦ 0
 18725 ¦  A Doctor in the House ¦  Mahathir Mohamad ¦  2011 ¦ 26
 15533 ¦  The Singapore Story ¦  Lee Kuan Yew ¦  1998 ¦ 20
 18929 ¦  A Brief History of Time ¦  Stephen Hawking ¦  1998 ¦ 25
 17911 ¦  Les Miserable ¦  Victor Hugo ¦  1862 ¦ 96
 24501 ¦  The Bourne Identity ¦  Robert Ludlum ¦  1980 ¦ 71
 14796 ¦  The Lion the Witch and the Wardrobe ¦  C.S. Lewis ¦  1950 ¦ 68
 18989 ¦  The Mist ¦  Stephen King ¦  1980 ¦ 9
 15263 ¦  Cosmos ¦  Carl Sagan ¦  1980 ¦ 20
 18677 ¦  Anna Karenina ¦  Leo Tolstoy ¦  1877 ¦ 136
30 row(s) read.

Process returned 0 (0x0)   execution time : 0.167 s
Press any key to continue.
```

Note:

- You are free to modify the codes given (BookItem and main application file) to suit your program. Important: You do not need to understand how readData() works, so please do not tamper with it if you are not familiar!
- This assignment does not require you to perform any update to the text file database.
- Feel free to add other methods to **SortedLList** class that will help you in task 2.
- Hard-coding the answer to the statistics is not allowed. Anyway, the instructor will be using a different set of book data during evaluation. Moreover, we will provide a larger set of data a few days before the deadline to help you test your program further.

Database text file



```
14833,Alice in Wonderland,Lewis Carroll,1865,28
15322,War of the World,H.G. Wells,1953,17
20983,The City,Dean Kootnz,2014,36
19123,The Hobbitt,J.R.R. Tolkien,1938,178
24504,The Old Man and the Sea,Ernest Hemingway,1951,135
13323,My Experiments with Truth,Mahatma M.K.Gandhi,1929,6
19232,Far from the Madding Crowd,Thomas Hardy,1874,2
11219,The Merchant of Venice,William Shakespeare,1605,26
15677,A Tale of Two Cities,Charles Dickens,1859,37
15899,David Copperfield,Charles Dickens,1850,29
```

db_small.txt contains information of the books that have been sold previously. Each row consists of the following book data, delimited by a comma (,):

`serial number, book title, author name, year published, copies sold`

If you are compiling at command prompt, this file should be placed in the same directory as your codes. However, if you are creating a CodeBlocks project, the database file should be placed in the same location as the project file (.cbp).

What do hand up

Although there are two separate tasks in this assignment, you are encouraged to combine both tasks into a single program (either showing one followed by the other, OR create a simple selection menu to provide an option of which to show).

Bonuses (Optional)

There are more challenging features that you can attempt if you have finished the required tasks above. Additional bonus marks (up to 3% but not exceeding your total coursework marks) will be awarded if you can add the following features to task 1:

- **SortedMerge** - Merges another sorted linked list (passed in as a parameter) into the linked list and ensure the merged list is in sorted order. No new linked list or new nodes should be created.
- **SortedIntersect** – Find the intersecting nodes between the linked list and and another sorted linked list (passed in as a parameter), and returns a linked list containing the intersecting nodes. By exploiting the fact that both linked lists are sorted, we only *traverse each list once*.

Submission Format

1. Create a zip file name

TT0X_AS2_StudentID_StudentName.zip

Modify the parts in read. *TT0X* is your **lab section** (replace the 'X' with the correct number). *StudentID* is your student ID. *StudentName* is your name. RAR or 7ZIP files are also acceptable. The zip should contain

a) **Source code files** (*.hpp and *.cpp) or **complete set of project files** (*.cbp, *.hpp, *.cpp, if you have used CodeBlocks Project). Do not attach any .exe file. One easy way is to delete away the generated .exe files before zipping up for submission.

b) **User manual** (*.txt) containing simple instructions outlining how to use your program, and what your program can do.

2. Compose an email with subject as "*TT0X StudentID StudentName*".
3. Attached the *TT0X_AS2_StudentID_StudentName.zip* created in Step 1.
4. Submit your assignment to the MMLS system.
5. The Submission will be regarded as late according to MMLS timestamps. Submit at least several hours earlier to be safe, in case you encounter any potential Internet problems at the wrong timing.

Feature Sheet & Evaluation Criteria

Criteria	Item
1. SortedLList (4 marks)	1.1. SortedInsert [1m]
	1.2. Retrieve [1m]
	1.3. Delete [1m]
	1.4. Basic Linked List ADT complete [1m]
2. Statistics (5 marks)	2.1. Successfully integrated SortedLList [1m]
	2.1. Book List (sorted by title) [1.5m]
	2.2. Book Search [1m]
	2.3. Top 10 best sellers [1.5m]
3. Miscellaneous (1 marks)	3.1. Good programming style, proper indentation [0.5m]
	3.2. Separate compilation [0.5m]
4. Bonus* (max. 3 marks to coursework mark)	4.1 SortedMerge [1.5m] 4.2 SortedIntersect [1.5m]
5. Interview** (0 mark for the assignment if fail to be present for interview)	Fluency in using the program
	Ability to explain code
6. Work found to have plagiarized from another source, or Late submission	0 mark for the whole assignment

Note:

* Bonuses are added on top of your total assignment mark of 10, and will be awarded based on the discretion of your lecturer. However, total coursework marks will be capped at 60 marks.

** Interview carries no mark (hence, no difference for students who are called for interview and those who are not). It is for the lecturers to ask questions if there are any doubts over your assignment, or if you are required to clarify parts of your work so that the lecturers understand your work better. However, it is compulsory to be present for interview if selected.