CSC 2110, Spring 2013
Programming Assignment 4: Zip Code Binary Search Tree
Assign date: April 05, 2013, Due: April 17, 2013 at 9:00pm

Skills you will learn

- Building a BST
- Searching a BST
- Deleting items from a BST
- Adding items to a BST

Introduction

Similar to your second programming assignment, in this assignment you will create an in-memory database. However, your database will load a text file of zip codes. **The name of the text file will be provided as a command line argument**. The text file contains a string zip code, the name of the city/region, and the state in separate lines. Your program should begin by reading the text file and creating a linked binary search tree of zip codes. Once the tree is created, your program will then prompt the user with a menu of operations. The menu will include operations for displaying a particular zip code, adding a zip code, removing a zip code, printing the entire BST of zip codes in sorted order, or exiting. Here is an example run:

```
$ ./zipLibrary zips10.txt
Loading library, please wait...
Welcome to the Zips Library!


What would you like to do?

(F)ind an article
(L)ist all articles
(A)dd a new article
(R)emove an existing article
(E)xit

Your choice>L
-----------------------------------------------------
  Zip Code                   City/Region     State
-----------------------------------------------------
     12016                   AURIESVILLE        NY
     15329                    PROSPERITY        PA
     28787                   WEAVERVILLE        NC
     38224                 COTTAGE GROVE        TN
     50168                         MINGO        IA
     62461                      SHUMWAY         IL
     62879                SAILOR SPRINGS        IL
     80274                        DENVER        CO
     89039                   CAL NEV ARI        NV
     91103                      PASADENA        CA
-----------------------------------------------------
10 total records shown.
```

```
What would you like to do?

(F)ind an article
(L)ist all articles
(A)dd a new article
(R)emove an existing article
(E)xit

Your choice>F

Please enter a search key: 62461

------------------------------------------------------------
- Zip Code: 62461
- City/Region: SHUMWAY
- State: IL
------------------------------------------------------------

What would you like to do?

(F)ind an article
(L)ist all articles
(A)dd a new article
(R)emove an existing article
(E)xit

Your choice>F

Please enter a search key: some nonexistent zipcode

------------------------------------------------------------
-   Sorry, but there are no records matching your query   -
------------------------------------------------------------

What would you like to do?

(F)ind an article
(L)ist all articles
(A)dd a new article
(R)emove an existing article
(E)xit

Your choice>A

Please enter the new zip code: 12016

------------------------------------------------------------
-                This key already exists!                  -
------------------------------------------------------------

What would you like to do?

(F)ind an article
(L)ist all articles
(A)dd a new article
(R)emove an existing article
(E)xit
```

```
Your choice>A

Please enter the new zip code: 38501
Enter the city/region: Cookeville
Enter the state: TN
------------------------------------------------------------
- The following record has been added:
-
- Zip Code: 38501
- City/Region: Cookeville
- State: TN
------------------------------------------------------------

What would you like to do?

(F)ind an article
(L)ist all articles
(A)dd a new article
(R)emove an existing article
(E)xit

Your choice>L
-----------------------------------------------------
  Zip Code                   City/Region     State
-----------------------------------------------------
     12016                   AURIESVILLE        NY
     15329                    PROSPERITY        PA
     28787                   WEAVERVILLE        NC
     38224                 COTTAGE GROVE        TN
     38501                    Cookeville        TN
     50168                         MINGO        IA
     62461                      SHUMWAY         IL
     62879                SAILOR SPRINGS        IL
     80274                        DENVER        CO
     89039                   CAL NEV ARI        NV
     91103                      PASADENA        CA
-----------------------------------------------------
11 total records shown.

What would you like to do?

(F)ind an article
(L)ist all articles
(A)dd a new article
(R)emove an existing article
(E)xit

Your choice>R
Please enter the zip code that you wish to remove: 123456

------------------------------------------------------------
-   Sorry, but there are no records matching your query   -
------------------------------------------------------------
```

```
What would you like to do?

(F)ind an article
(L)ist all articles
(A)dd a new article
(R)emove an existing article
(E)xit

Your choice>R
Please enter the zip code that you wish to remove: 38501
-----------------------------------------------------------
- The following record has been REMOVED:
-
- Zip Code: 38501
-----------------------------------------------------------

What would you like to do?

(F)ind an article
(L)ist all articles
(A)dd a new article
(R)emove an existing article
(E)xit

Your choice>asdf jkl;
Invalid choice.  Please choose again>E
Thank you for using the Zips Library!
```

If you look carefully at the output above, you will notice that the file is completely loaded when the program starts up. In other words, your program should read the file and create a BST of all the zip codes immediately. Reading the file is simple. Each zip consists of three lines. The first line is the unique zip code, the second line is the city/region, and the third line is the state code. Following is an example:

```
28787
WEAVERVILLE
NC
89039
CAL NEV ARI
NV
12016
AURIESVILLE
NY
15329
PROSPERITY
PA
91103
PASADENA
CA
38224
COTTAGE GROVE
TN
```

To process all entries in the file, simply read each triplet of lines, insert a new node into your BST, and then iterate (loop again). After the BST is constructed, your program prints out the menu and waits for the user's input. Selecting 'F' on the menu (hitting 'F' on the keyboard and then <Enter>) results in the user being prompted for a zip code. The zip code is a string of 5 digits. Once the user enters a zip code, your program should search for the zip code in your BST, and, once the zip code is found, prints out the corresponding zip code, city, and state. If the zip code is not found, your program should give the appropriate message.

Selecting 'L' lists all zip codes in your BST. In other words, your program will traverse the BST, in-order, printing out the zip code, city, and state stored in each node. After printing out the information for each node in your BST, the total number of entries should be printed.

Selecting 'A' adds a new zip code to the list. To add a new zip code, your program must first prompt the user for the zip code, city, and state. Note that once the user enters the zip code, your program should check to see if that key already exists and, if so, inform the user that that key already exists. Once the user enters the zip code's data, your program should put the item into the BST, making sure that the tree remains a BST (recall the properties of a BST: for each node, all nodes in the left subtree are less than the node and all nodes in the right subtree are greater than the node).

Selecting 'R' removes an existing zip code from the BST. To remove a zip code, your program must first prompt the user for the zip code. Once the zip code is entered, your program searches for it, and if found, removes that node from the BST and deallocates its memory. If the zip code is not found in the BST, your program should print out the corresponding message that the zip code was not found. Lastly, selecting 'E' exits the program.

Implementation:

- You must implement a Binary Search Tree class.
- The BST class that you implement must be composed of instances of a Zipcode object (Each node contains a Zipcode object as well as pointers to that node's left and right children). The Zipcode class must contain private data members for the zipcode, city/region, and state values for each entry in the supplied text files (You may have to overload some comparison operators too).

Deliverables:

- You must include a README file describing what your program does and how to compile and run your program.
- You need to zip all your source code (all header and cpp files only, no executable or data files) and the README file and submit it to iLearn. The name of your zip file should be lastname_prog4_2110.zip (for example my filename would be mcdaniel_prog4_2110.zip.

Your program will not be graded if it does not compile

Remember to #include <cstdlib> so that there will be no problem compiling your program on Windows machines.

Grading:

This program is worth 100 points, distributed as follows:

Functionality (program works correctly, and meets the specifications)	**85 pts**
Program Style	**10 pts**
Meaningful variable names	
Proper indentation	
Correct program header	
Comments throughout the program	
README file	**5 pts**

** If your program doesn't compile your score for the assignment will be zero.

Late submission:

There will be a 5% penalty/day for up to two days. For delays more than two days a score of zero will be assigned.

Start working on your assignment early and don't hesitate to contact the TA or me if you need help.