CSCI 2270 – Data Structures and Algorithms
Instructor: Hoenigman
Assignment 6
Due Wednesday, March 2 by 3pm

Binary Search Trees

An online movie service needs help keeping track of their stock. You should help them by developing a program that stores the movies in a Binary Search Tree (BST) ordered by movie title. For each of the movies in the store's inventory, the following information is kept:

- IMDB ranking
- Title
- Year released
- Quantity in stock

Your program will have a menu similar to assignments 3, 4, and 5 from which the user could select options. In this assignment, your menu needs to include options for finding a movie, renting a movie, printing the inventory, and quitting the program.

Note: Assignment 7 will build on this assignment. If you don't complete Assignment 6, you will be behind on Assignment 7.

Your program needs to incorporate the following functionality.

Insert all the movies in the tree.
When the user starts the program they will pass it the name of the text file that contains all movie information as a command line argument. There is a file on Moodle called Assignment6Movies.txt that you should use for this assignment. Your program needs to open the file, and read all movie data in the file. From this data, build the BST ordered alphabetically by movie title. All other information about the movie should also be included in the node in the tree. *Note: the data should be added to the tree in the order it is read in.*

Helpful hint: There is a string compare function that makes it easy to determine if one string is less than another, which can be used to compare two strings alphabetically. For more information, look here: *http://www.cplusplus.com/reference/string/string/compare/*.

After the tree has been built, display a menu with the following options.

Menu Options:

1. **Find a movie.**

When the user selects this option from the menu, they should be prompted for the name of the movie. You program should then search the tree and display all information for that movie. If the movie is not found in the tree, your program should display, “Movie not found.” The findMovie method included in the MovieTree.h header file is a void type. All movie information should be displayed in the findMovie method.

2. **Rent a movie.**
   When the user selects this option from the menu, they should be prompted for the name of the movie. If the movie is found in the tree, your program should update the Quantity in stock property of the movie and display the new information about the movie. If the movie is not found, your program should display, “Movie not found.” If the movie is found in the tree, but the Quantity is zero, display “Movie out of stock.”. Just like findMovie, rentMovie is also a void type. Information about the movie rented should be displayed in the rentMovie method.

3. **Print the entire inventory.**
   When the user selects this option from the menu, your program should display all movie titles and the quantity available in sorted order by title. See the lecture notes on in-order tree traversal and your textbook for more information.

4. **Quit the program.**
   When the user selects this option, your program should exit.

Implementation details
Your BST should be implemented in a class. You are provided with a MovieTree.h file on Moodle and you need to implement the corresponding MovieTree.cpp file and Assignment6.cpp file. The movie data is contained in a file called Assignment6Movies.txt.

To submit your work, zip all files together and submit them to COG as Assignment6.zip. If you do not get your assignment working on COG, you will have the option of a grading interview.

Appendix A – cout statements that COG expects

Display menu

```
cout << "======Main Menu======" << endl;
cout << "1. Find a movie" << endl;
cout << "2. Rent a movie" << endl;
cout << "3. Print the inventory" << endl;
cout << "4. Quit" << endl;
```

Find a movie

```
cout << "Enter title:" << endl;
```

Display found movie information

```
cout << "Movie Info:" << endl;
cout << "===========" << endl;
cout << "Ranking:" << foundMovie->ranking << endl;
cout << "Title:" << foundMovie->title << endl;
cout << "Year:" << foundMovie->year << endl;
cout << "Quantity:" << foundMovie->quantity << endl;
```

If movie not found

```
cout << "Movie not found." << endl;
```

Rent a movie

```
//If movie is in stock
cout << "Movie has been rented." << endl;
cout << "Movie Info:" << endl;
cout << "===========" << endl;
cout << "Ranking:" << foundMovie->ranking << endl;
cout << "Title:" << foundMovie->title << endl;
cout << "Year:" << foundMovie->year << endl;
cout << "Quantity:" << foundMovie->quantity << endl;
//If movie is out of stock
cout << "Movie out of stock." << endl;

//If movie not found in tree
cout << "Movie not found." << endl;
```

Print the inventory

```
//For all movies in tree
cout<<"Movie: "<<node->title<<" "<<node->quantity<<endl;
```

Quit

```
cout << "Goodbye!" << endl;
```