WAYNE STATE UNIVERSITY
COLLEGE OF ENGINEERING

Computer Science Department

CSC 2201: Computer Science II – Lab
Lab9

Description:

You will implement Binary Search Tree using linked list.

Goals:

Learn how to implement Binary Search Tree ADT with linked data structures using C++ pointers.

Book Reference:

Please read chapter 9 on the lab book carefully. Especially pages 109-111 of the book.

Where to Start:

1. Download Lab9_Work.zip from Blackboard
2. Unzip the file
3. Open the solution in Microsoft Visual Studio
4. Make sure that the project does not show any compile errors.
5. Implement methods and operations of the QueueArray class:

5.1 Implement `BSTreeNode::BSTreeNode`

```
// Creates a binary search tree node containing data item elem, left
// child pointer leftPtr, and right child pointer rightPtr.
```

5.2 Implement `BSTree<DataType, KeyType>::BSTree ()`

```
// Creates an empty tree.
```

5.3 Implement `BSTree<DataType, KeyType>::BSTree ( const BSTree<DataType,KeyType>& other )`

```
// Copy constructor.
```

5.4 Implement `operator =`

```
// Sets a tree to be equivalent to the tree "other".
```

5.5 Implement `~BSTree ()`

```
// Frees the memory used by a tree.
```

5.6 Implement `insert(newDataItem)`

```
// Inserts newDataItem into a tree. If an data item with the same
key
```

```
// as newDataItem already exists in the tree, then updates that
// data item's data with newDataItem's data.
```

5.7 Implement `retrieve (searchKey, searchDataItem )`

```
// Searches a tree for the data item with key searchKey. If the data
item
// is found, then copies the data item to searchDataItem and returns
true.
// Otherwise, returns false with searchDataItem undefined.
```

5.8 Implement `remove (deleteKey )`

```
// Removes the data item with key deleteKey from a tree. If the
// data item is found, then deletes it from the tree and returns
true.
// Otherwise, returns false.
```

5.9 Implement `writeKeys ()`

```
// Outputs the keys in a tree in ascending order.
```

5.10 Implement `isEmpty ()`

```
// Returns true if a tree is empty. Otherwise returns false.
```

5.11 Implement `clear ()`

```
// Removes all the nodes from a tree.
```

Activate `LAB9_TEST1` and `LAB9_TEST2` and `LAB9_TEST3` in the config.h file.

5.12 Implement `getCount ()`

```
// Returns the number of nodes in the tree
```

5.13 Implement `getHeight ()`

```
// Returns the height of a tree.
```

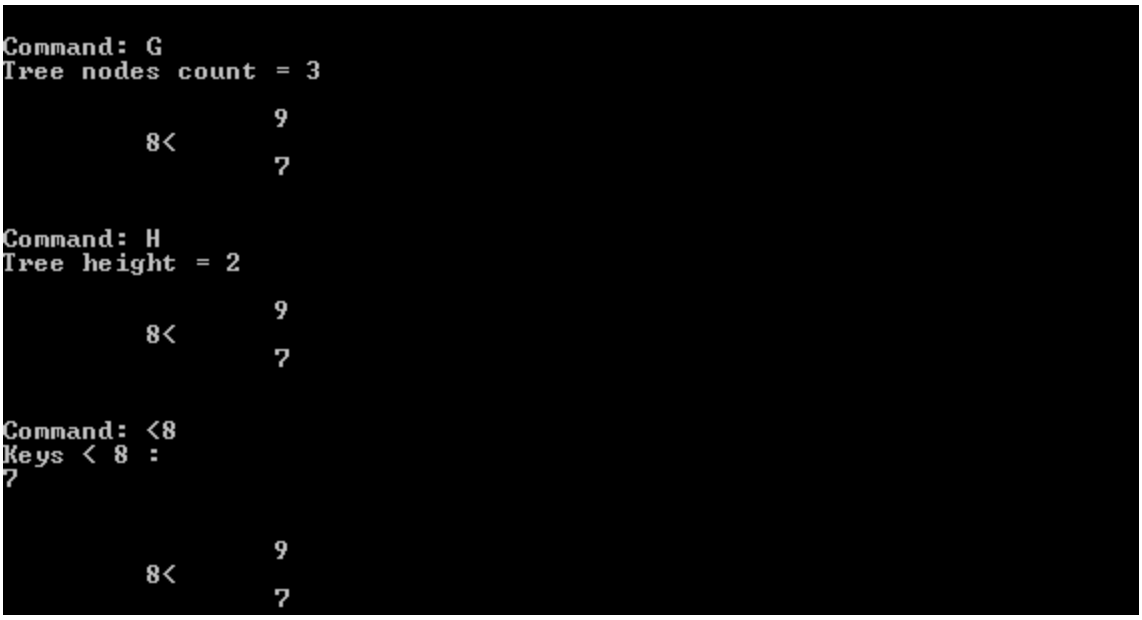
5.14 Implement `writeLessThan (searchKey)`

```
// Outputs the keys in a tree that are less than searchKey.
```

6. Test your implementation using the program in the file test9.cpp.

7. Output:

```
Commands:
  P    : [P]rint Help (displays this message)
  +key : Insert (or update) data item (use integers)
  ?key : Retrieve data item
  -key : Remove data item
  K    : Write keys in ascending order
  C    : Clear the tree
  E    : Empty tree?
  G    : Get count of nodes          (Inactive : In-lab Exercise 2)
  H    : Height                      (Inactive : In-lab Exercise 2)
  <key : Write keys that are < key   (Inactive : In-lab Exercise 3)
  Q    : Quit the test program

Empty tree

Command: +8
Insert : key = 8

         8


Command: +9
Insert : key = 9

                9
         8/


Command: +7
Insert : key = 7

                9
         8<
                7


Command: ?8
Retrieved : getKey = 8

                9
         8<
                7


Command: ?6
Not found

                9
         8<
                7


Command: K
Keys:
7 8 9

                9
         8<
                7


Command: -7
Removed data item

                9
         8/


Command: E
Tree is NOT empty

                9
         8/
```



Create a Zip file of your solution:

1. Right click on your solution in Solution Explorer
2. Click on "Open Folder in File Explorer"
3. Go one level up in file explorer
4. Right click on your solution folder
5. Add it to archive by creating a zip file

Upload the zipped file on Blackboard:

1. Go to Blackboard
2. Click on this course (CSC 2201: Computer Science II Lab)
3. Go to the folder "Labs"
4. Click on the "Lab9_Work" assignment
5. Upload your zipped file