Lab 5

Binary Trees

Due 12/07/07

In this lab you will implement an ordered Binary Tree (also called a Binary search tree) as a linked ADT. Each node in the tree will contain an integer value, and pointers to a left and right subtree. In an ordered binary tree, each node in the left subtree of a node has a value less than the value stored in the "root" of the subtree, and every node in the right subtree has a value greater than the value stored in the "root" of the subtree. To complete this project you will:

1) Create a binaryTree ADT, containing a constructor, an inner Node class, and instance methods to:
   - **Add**: this will add a new value into the binary tree. It should receive a parameter an integer value to be inserted into the tree. If the value is already stored in some node in the tree you should print an error message.
   - **Search:** This function will receive a single integer value and will search the binary tree to determine if the value is present in the tree. If the value is found a message should be printed that the value is in the tree, otherwise an error should be displayed
   - **Delete:** This function will receive a single integer value and then locate the node containing that value and then deletes it from the tree. When we delete a node from the tree, we replace it with the left most leaf node of the deleted nodes right subtree.
   - **Height:** This method calculates the height of a binary tree.
   - **Inorder:** prints an inorder traversal of the binary tree
   - **Preorder:** prints a preorder traversal of the binary tree
   - **Postorder:** prints a postorder traversal of the binary tree
   - **Count_nodes:** counts the number of nodes (values) stored in the binary tree.
   - **Ancestors:** given an integer value, this (recursive) function prints out the ancestors of the node (containing the entered value), in reverse order.

2) In addition to your ADT, You will create a main program which will provide the user with a menu of options which will call functions provided in your ADT

FOR MORE INFORMATION ON TERMINOLOGY CONCERNING BINARY TREES OR ON RECURSIVE FUNCTIONS, See chapter 11 of your text .