

Class CMPS 261
Section 001
Problem Programming Assignment #1
Name McKelvy, James Markus
CLID Jmm0468
Due Date 12:30pm September 22, 2005

I. Requirements Documentation

I.1 Description of the Problem

Name: Generic Templated Binary Tree

Problem Statement: A generic templated binary tree needs to be implemented, with functions that allow inorder, preorder, and postorder traversal.

Problem Specification: The program needs to be able to create an empty binary tree, a binary tree with only one node and two empty subtrees, and a binary tree with a root node and two (possibly) non-empty subtrees. Also the binary tree needs to be able to tell if it is empty or not, meaning, is there data stored in the nodes. The binary tree must also have its own methods of traversal using inorder, preorder, and postorder traversal. There will be no input into the binary tree aside from the initial creation of the binary tree. The only output should be from a visit function that simply reveals the contents of its root node.

I.2 Input Information

I.2.1 Input Streams

Name: N/A

Description: N/A

Format: N/A

Size: N/A

Sample: N/A

I.2.2 Input Items

Description: N/A

Type: N/A

Range of acceptable values: N/A

I.3 Output Information

I.3.1 Output Streams

Name: cout

Description: Used to output the testing of the BinaryTree

Format: Displays what is currently happening such as "Creating a binary tree" in addition to (possibly) output of an emptiness test, as well as an inorder, preorder, and postorder,

traversal each with their own outputs, and lastly, a line waiting for user input (a pause function)

Size: At least 10 lines

Sample:

Creating an empty binary tree.

Is tree1 empty? Yes.

Performing inorder traversal.

Performing preorder traversal.

Performing postorder traversal.

Press ENTER.

I.3.2 Output Items

Description: N/A

Type: N/A

Range of acceptable values: N/A

I.4 User Interface Information

I.4.1 Description

The program will not allow the user many options. The program will run without a menu and simply test the integrity of the implementation of a binary tree. The program will create an empty tree and use the function isEmpty() to test whether the tree is truly empty, as well as perform inorder, preorder, and postorder traversals and output the information the user. The program will pause for the user. Next the program will create a non-empty tree and use the function isEmpty() to test where the tree is truly non-empty. The program will pause for the user. Next the program will create a non-empty tree and with only one node and two empty subtrees, and output the inorder, preorder, and postorder traversal results to the user. The program will pause for the user. Next the program will create a binary tree with a root node and two non-empty subtrees. The program will output the inorder, preorder, and postorder traversal results to the user. The program will pause before exiting.