

Programming Assignment

Binary Search Tree

Due Date : Friday April 30th , 2021
No Later Than @ 5:00 pm

Using the BST program that appeared in my notes ,
modify existing functions and adding new following
functions as follows :

1. Construct a BST that consist of 6 – 9 nodes instead of 5. Use the same nodes that appears in the example or any other values
2. Display the binary search tree as it shown in the notes
3. Write a function that traverse and displays the tree in pre-order form.
4. Write a function that traverse and displays the tree in post -order form.
5. Write a function that counts and displays the number of leafs in the tree.
6. Write a function that displays only the leafs of the tree.
7. Write a function that searches for an element in the tree.
8. Write a function that deletes an element from the tree.

Notes :

- **This assignment will replace the lowest programming assignment grade**
- Must use and modify the program that was discussed in the class and appears in my notes. You are not allowed to eliminate any functions in the program.
- Also, must use the main program that is provided in the sample run.
- You have to use **only arrays** when implementing this program. Cannot use vector arrays , or linked lists.
- Cannot use library functions such as swap, sort , vector ... etc. for this programming assignment.
- Can not use any global variables , arrays ... etc.

Guidelines:

At the beginning of your program (and **before** the #include statement), include the following :

Header comments (file documentation block) should be at the top of each file and should contain: Author / s, Due Date, Assignment Number, Course number and section, Instructor, and a brief description of the purpose of the code in the file. For example :

```
//  
//  Author : (Your name here!!)  
//  
//  Due Date :  
//  Optional Program .  
//  
//  Spring 2021 - CS 3358 - Your Section Number  
//  
//  Instructor:  Husain Gholoom.  
//  
//  <Brief description of the purpose of the program>
```

Variable names :

- Must be meaningful.
- The initial letter should be lowercase, following words should be capitalized, no other caps or punctuation (i.e. weightInPounds).
- Each variable must be declared on a separate line with a descriptive comment.

Named constants :

- Use for most numeric literals.
- All capitals with underscores (i.e. TX_STATE_SALES_TAX)
- Should occur at top of function, or global (only if necessary)

Line length of source code should be no longer than 80 characters (no wrapping of lines).

Indentation :

- Use 2-4 spaces (but be consistent throughout your program).
- Indent blocks, within blocks, etc.
- Use blank lines to separate sections.

Comments for variables :

All variable definitions should be commented as follows:

```
int gender; // integer value for the gender,  
           // 1 = Male , 2 = Female ,
```

Rules :

1. **This assignment will replace the lowest programming assignment grade.**
2. Your program **must compile** and run. We will test your program using the latest version of Codeblock which is 20:03. (No Global Compiler Special Flags are set. You are not allowed to use C++11, C++14 ... etc.) .
3. Your program should run for any number of nodes.
4. Your program must be **documented according to the style above . See the website for the sample programming style program.**
5. Must properly format the output as shown below.
6. Must use the program that appears in my notes. The program will be modified and new functions are added.
7. You must name your program as :

- **LastName_FirstName_P7_BST.cpp**

Where LastName is your Last Name and FirstName is your First Name. For example , the file name should look something like :

Gholoom_Husain_P7_BST.cpp (not .cbp)

8. Must upload the electronic version of the program no later than 5:00 pm on the due date. **DO NOT send your assignment solution via email. No late assignments will be accepted and a grade of zero will be assigned .**

The Maximum Grade for this assignment is 10 / 10.

The following points will be deducted if :

- Compilation errors , Incorrect file format such as uploading .cbp instead of .cpp , missing electronic copy , not using the provided program that is in my lecture notes, using vector array or linked lists , using function libraries such as swap / sort (- 10 points)
- Each logical error : (at least - 1 points)
- Other : (at least 1.25 points) if any of the following takes a place :
 - Incorrect program file name.
 - Incorrect output format.
 - Incorrect Style **such as but not limited to** Missing title , footer , comments, or program documentations , missing or incorrect section number, not replacing my name with your name in the footer.

Use the following .cpp file as the main of your program

```
BinarySearchTree    tree(30);

cout << "Binary Search Tree by Husain Gholoom.\n\n";

cout << "Inserting Nodes.\n\n";

tree.insertElement(4);
tree.insertElement(6);
tree.insertElement(9);
tree.insertElement(3);
tree.insertElement(2);
tree.insertElement(8);

cout << "Building BST is completed.\n\n";

// Pre-Order Traversal .
cout << "Pre-Order Traversal of the BST :\n\n";
tree.preOrder();

// Post-Order Traversal .
cout << "Post-Order Traversal of the BST :\n\n";
tree.postOrder();

// Counting Number of Leafs .
cout << "\n\nNumber of Leafs =      " <<
tree.treeLeafsCount();
```

```
// Display leaf values of BST
cout << "\n\nHere are the leaf values in the BST:\n\n";
tree.displayLeafValues();

// Searching for an element in BST
cout << "\n\nSearching for 9 in the BST:\n\n";
tree.searchElement(9);

// Searching for an element in BST
cout << "\n\nSearching for 5 in the BST:\n\n";
tree.searchElement(5);

// Deleting an element
cout << "\n\nDeleting 7 from the BST:\n\n";
tree.deleteElement(7);

// Deleting an element
cout << "\n\nDeleting 2 from the BST:\n\n";
tree.deleteElement(2);

// Display final tree
cout << "\n\nDisplaying final BST:\n\n";

cout << "\n\n\nHusain Gholoom - April 30 , 2021\n\n";
```


Sample Output Run

Binary Search Tree By Husain Gholoom

Inserting Nodes.

Adding: 4 Inserted at index: 0
Adding: 6 Right - Inserted at index: 2
Adding: 9 Right - Inserted at index: 6
Adding: 3 Left - Inserted at index: 1
Adding: 2 Left - Inserted at index: 3
Adding: 8 Left - Inserted at index: 13

Building BST is completed.

Pre-Order Traversal of the BST :

4
3
2
6
9
8

Post-Order Traversal of the BST :

2
3
8
9
6
4

Number of Leafs = 2

Here are the leaf values in the BST:

2

8

Searching for 9

9 found at index 6

Searching for 5

5 is not found

Deleting 7

7 is not found

Deleting 2

2 is deleted

Final Tree

4 at index: 0

6 at index: 2

9 at index: 6

3 at index: 1

8 at index: 13

Husain Gholoom - April 30 , 2021