DTS Lab 6  
binary search tree

Contents

[Objective 2](#_Toc358034623)

[Prototypes 2](#_Toc358034624)

[Desired Output 3](#_Toc358034625)

[Submission 5](#_Toc358034626)

# Objective

Implement a Binary Search Tree template class *BST*.

Place all your code in a file named *BST.h*

# Data Members

The BST class will have the following private data members:

Node\* root = nullptr;

# Prototypes

The BST class will have the following public interface:

/////////////////////////////////////////////////////////////////////////////  
// Function : Constuctor  
// Notes : does nothing  
/////////////////////////////////////////////////////////////////////////////  
BST()  
  
/////////////////////////////////////////////////////////////////////////////  
// Function : Destructor  
// Notes : deletes each node in the tree recursively  
/////////////////////////////////////////////////////////////////////////////  
~BST()  
  
/////////////////////////////////////////////////////////////////////////////  
// Function : assignment operator  
// Notes: Deep copy of that tree. Starting at that root, recursively  
// call insert in a helper function  
/////////////////////////////////////////////////////////////////////////////  
BST& operator=(const BST& that);  
  
/////////////////////////////////////////////////////////////////////////////  
// Function: copy constructor  
// Notes: call assignment operator  
/////////////////////////////////////////////////////////////////////////////  
BST(const BST& that);  
  
/////////////////////////////////////////////////////////////////////////////  
// Function : insert  
// Parameters : v - the item to insert into the tree   
/////////////////////////////////////////////////////////////////////////////  
void insert(const Type& v)  
  
/////////////////////////////////////////////////////////////////////////////  
// Function : findAndRemove  
// Parameters : v - the item to find (and remove if it is found)  
// Return : bool - true if the item was removed, false otherwise  
// Notes: Find the correct node first, then children cases:  
// A) 2 children ("fix" tree)  
// B) 0 children  
// C) 1 children  
/////////////////////////////////////////////////////////////////////////////  
bool findAndRemove(const Type& v)

/////////////////////////////////////////////////////////////////////////////  
// Function : find  
// Parameters : v - the item to find  
// Return : bool - true if the item was found, false otherwise  
// Notes: Does not need to be recursive  
/////////////////////////////////////////////////////////////////////////////  
bool find(const Type& v) const  
  
  
/////////////////////////////////////////////////////////////////////////////  
// Function : clear  
// Notes : Make helper function to recursively delete each node in a  
// post-order  
/////////////////////////////////////////////////////////////////////////////  
void clear()  
  
/////////////////////////////////////////////////////////////////////////////  
// Function : printInOrder  
// Notes : prints the contents of the BST to the screen, in ascending order  
// using resursive helper function  
/////////////////////////////////////////////////////////////////////////////  
void printInOrder() const

# Desired Output

Compile and run your code with the DTSLab6.cpp file provided via FSO. Your console output should match the following block identically:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\* LAB 6: \*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
\*\*\* TEST 1 \*\*\*  
20 25 30 35 40 45 50 55 60 65 70 75 77 80 85   
  
  
\*\*\* TEST 2 \*\*\*  
20 30 35 40 45 50 55 60 70 77 80 85   
The value 25 was not found..  
The value 30 was found!  
The value 35 was found!  
The value 40 was found!  
The value 45 was found!  
The value 50 was found!  
The value 55 was found!  
The value 60 was found!  
The value 65 was not found..  
The value 70 was found!  
The value 75 was not found..  
The value 80 was found!  
The value 85 was found!

\*\*\* TEST 3 \*\*\*  
20 25 30 35 40 45 50 55 60 65 70 75 77 80 85   
20 25 30 35 40 45 50 55 60 65 70 75 77 80 85   
20 25 30 35 40 45 50 55 60 65 70 75 77 80 85   
20 25 30 35 40 45 50 55 60 65 70 75 77 80 85   
25 30 35 40 45 50 55 60 65 70 75 77 80 85   
25 30 35 40 45 50 55 60 65 75 77 80 85   
25 30 35 40 45 50 60 65 75 77 80 85   
25 30 35 40 45 50 60 65 75 80 85   
25 30 35 40 45 60 65 75 80 85   
25 35 40 45 60 65 75 80 85   
35 40 45 60 65 75 80 85   
40 45 60 65 75 80 85   
40 45 60 75 80 85   
40 45 60 75 80   
40 45 60 75   
45 60 75   
60 75   
75   
  
  
\*\*\* TEST 4 \*\*\*  
4a:  
a b c d e f g h i j k l m n o   
4b:  
a b c d e f g h i j k l m n o   
4c:  
a b c d e f g h i j k l m n o   
4d:  
  
4e:

4e:

# Submission

To submit the lab assignment:

1. Clean, build, and run Lab6.cpp with your BST.h file in Visual Studio (debug mode).
   1. clear up any warnings you encounter.
   2. verify that your output is correct by comparing it to the lab document's Desired Output section, line-by-line.
   3. ensure there are no memory leaks.

2) Submit the "BST.h" file via FSO.