**DTS Lab 6  
*binary search tree***

**Objective**

Implement a Binary Search Tree template class *BST*.

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

**Prototypes**

The BST class will have the following public interface:

/////////////////////////////////////////////////////////////////////////////  
// Function : Constuctor  
// Notes : constucts an empty BST  
/////////////////////////////////////////////////////////////////////////////  
BST()  
  
/////////////////////////////////////////////////////////////////////////////  
// Function : Destructor  
// Notes : destroys the BST cleaning up any dynamic memory  
/////////////////////////////////////////////////////////////////////////////  
~BST()  
  
/////////////////////////////////////////////////////////////////////////////  
// Function : assignment operator  
/////////////////////////////////////////////////////////////////////////////  
BST& operator=(const BST& that);  
  
/////////////////////////////////////////////////////////////////////////////  
// Function: copy constructor  
/////////////////////////////////////////////////////////////////////////////  
BST(const BST& that);  
  
/////////////////////////////////////////////////////////////////////////////  
// Function : insert  
// Parameters : v - the item to insert   
/////////////////////////////////////////////////////////////////////////////  
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  
/////////////////////////////////////////////////////////////////////////////  
bool findAndRemove(const Type& v)

/////////////////////////////////////////////////////////////////////////////  
// Function : find  
// Parameters : v - the item to find  
// Return : bool - true if the item was found, false otherwise  
/////////////////////////////////////////////////////////////////////////////  
bool find(const Type& v) const  
  
  
/////////////////////////////////////////////////////////////////////////////  
// Function : clear  
// Notes : clears the BST, cleaning up any dynamic memory  
/////////////////////////////////////////////////////////////////////////////  
void clear()  
  
/////////////////////////////////////////////////////////////////////////////  
// Function : printInOrder  
// Notes : prints the contents of the BST to the screen, in ascending order  
/////////////////////////////////////////////////////////////////////////////  
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:

* Clean, build, and run DTSLab6.cpp with your BST.h file in Visual Studio (debug mode).
* clear up any warnings you encounter.
* verify that your output is correct by comparing it to the lab document's Desired Output section, line-by-line.
* ensure there are no memory leaks.
* On your desktop, create a new folder with your name in the following format:
* your last name
* a comma
* a single space
* your first name  
  **\* Appropriate capitalization for proper names should be used.**  
  Suitable examples include : "Pollack, Joey"; "De La Paz, Christhian"; "Tjarks, Matthew".
* Copy your 'BST.h' file into the folder that you created in step 2.. This is the only file I need and should therefore be the only file you submit.
* Right-click on the folder and select 'send to->compressed (zipped) folder'.
* Submit the compressed folder via FSO.