CS 2420 sections 001 and 003

AVL Trees

The standard template library is not to be used in this assignment or any other assignment unless expressly stated.

AVL Trees

Implement an **AVLTree** class that takes an integer as a data type. Including the following functions:

- Constructor
- Destructor
- Insert
- Print (indented print function like the one we did in class, but with each node's height)
- PreOrderTraversal

plus all other functions needed to properly implement an AVL balanced BST.

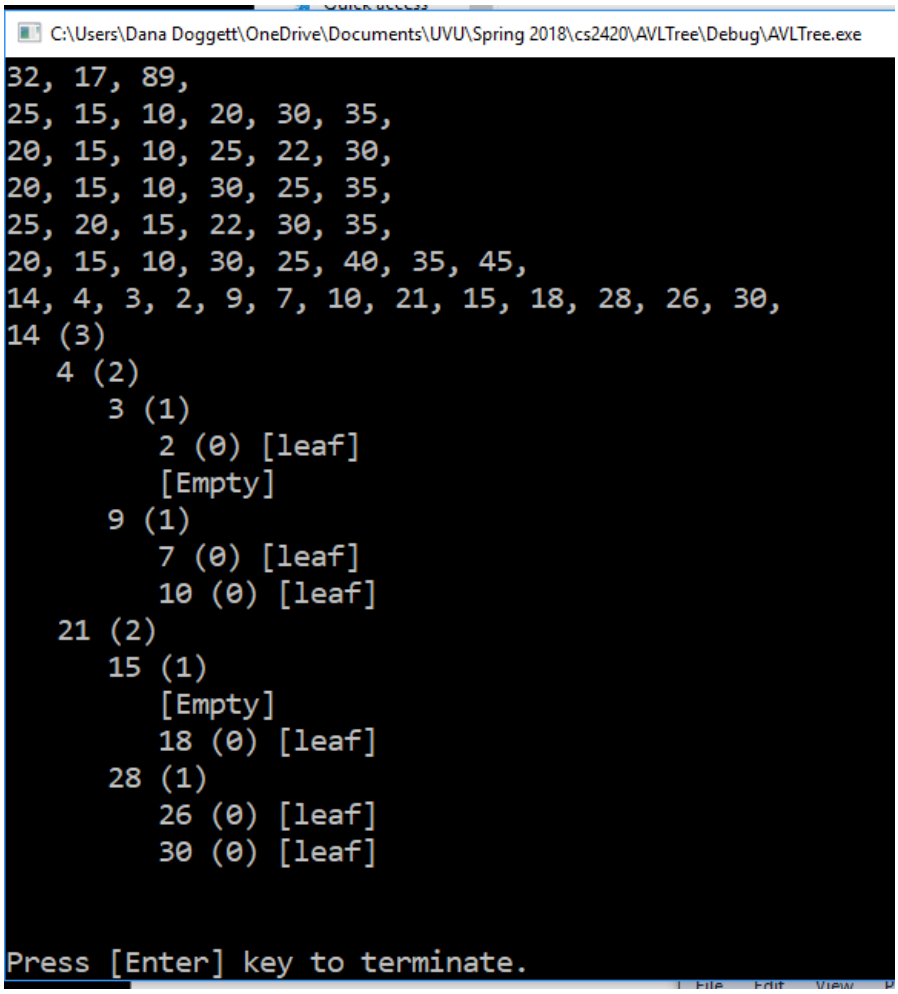
Also implement a **Node** class appropriate with this assignment.

Files To Create (and submit)

- AVLTree.cpp
- AVLTree.h
- Node.cpp
- Node.h

Testing

- The supplied Main.cpp should properly compile and run when added to your 4 files. You must not modify Main.cpp to get it to compile and run. When main() runs will create several AVLTrees and add data to each. After adding data, the tree will be traversed (with PreOrderTraversal()). When the final tree is traversed, it will also print that tree.
- The output should look like this:



Note

- Start Early!
- The rebalance function looks straight-forward, but can be tricky to implement correctly.
- You need to update height after each insert and after each rotate.
    - You should not adjust the height for every node in the tree, just those nodes visited in the insert function.

Grade Sheet
AVL Trees (115 pts)

Documentation (5 pts)

Student name, Section and Disclaimer (in AVLTree.cpp) _____/5

Trees (110 pts)

Constructor(s)	_____/5
Destructor	_____/5
Print	_____/10
Insert	_____/10
PreOrderTraversal	_____/10
Proper Output (10 pts for each data set) `	_____/70