**Individual Assignment**

**Extra Credit Assignment-Data Structures (2028C)**

***Topics covered: Working with Binary Search Trees***

**Objective:**

The objective of this homework is to create an implementation of a Binary Search Tree.

**Requirements:**

1. Create a templated Binary Search Tree class with the following methods. You may need or find it easier to create private methods and variable to simplify or eliminate duplicate code in the methods below such as RotateRight, RotateLeft or NodeHeight.
   1. **Constructor**
   2. **Destructor**
   3. **Insert** – accepts a value, creates a node, and inserts the node into the tree in the appropriate location. This should rebalance the tree as necessary, so the tree always remains balanced. If the value already exists in the tree, this should throw an error.
   4. **Find** – accepts a value, locates the value in the tree and returns a pointer to the node. If the value isn’t in the tree, it will return a null pointer.
   5. **Size** – returns the number of elements in the tree as an integer.
   6. GetAllAscending – returns an array with each node stored in order from smallest to largest (based on the sorting value, not the other data in the node).
   7. GetAllDescending – returns an array with each node stored in order from largest to smallest (based on the sorting value, not the other data in the node).
   8. EmptyTree – removes all nodes in the tree in a way to avoid memory leaks.
   9. **Remove** – accepts a value, finds the value, and removes it from the tree. A pointer to the removed node is returned. This should rebalance the tree as necessary, so the tree always remains balanced.

**Submission:**

Submit all source code files and any required data files in a zip file. Include a write up as a PDF including:

* Instructions for compiling and running the program including any files or folders that must exist.

Submission should be submitted via Canvas.

**Grading:**

1. 2.5 Points - BinarySearchTree class functions Insert, Find and Remove work correctly (excepting for keeping the tree balanced) including avoiding memory leaks.
2. 2.5 Points - BinarySearchTree stays always balanced with a difference of the longest path to a leaf and the shortest path to a leaf no greater than 2.
3. 5 Points - The other required functions of the BinarySearchTree work correctly including avoiding memory leaks.

The Code should be well formatted, well commented and follows a reasonable style. If program fails to compile, the grade will be 0 Points.