# CptS 233 Micro Assignment #4

For this micro assignment, you must implement the following AVL tree functions found inside AvlTree.java:

## Balance

This function determines where the imbalance at root exists (either right child or left child) and calls the appropriate rotation function (rotateLeft / rotateRight). The AvlNode class contains a getBalanceFactor() function that returns the balance factor for the node.

## RotateLeft / RotateRight

These functions rotate the supplied root pointer either left (rotateLeft) or right (rotateRight). Be sure to use the following rotation algorithms:

### Left (counter-clockwise) Rotation

1. Let CurrentRoot = the original root
2. Let NewRoot = CurrentRoot's **right** child
3. Set CurrentRoot's **right** child = NewRoot's **left** child
4. Set NewRoot's **left** child = CurrentRoot

### Right (clockwise) Rotation

1. Let CurrentRoot = the original root
2. Let NewRoot = CurrentRoot's **left** child
3. Set CurrentRoot's **left** child = NewRoot's **right** child
4. Set NewRoot's **right** child = CurrentRoot

## Grading

Your submission will be graded based on the following:

1. [10] Your solution builds, does not cause any runtime issues, and passes all test cases
2. [5] Follow the example of how pre-order traversal is done, implement your own in-order traversal in the same way.
3. [5] Write up some test cases of your own that shows your function works correctly.

Submit the hash value (first 7 characters is enough) of your final commit (before the due time) to this dropbox - no need to submit the code here. I will use that specific commit for grading.