CSCI 230, Summer 2015
HW 4 Queue ADT implemented as doubly-linked list

Collaboration policy: the assignment is to be completed individually. You should write these methods yourself, based on those provided in your text. You MAY NOT use any online resources. **Be sure to site all resources** that you use. You may discuss the assignment with the instructor or with the TA. You may post general questions to piazza, and specific questions to the instructors privately on piazza. You may not post code publically.

The AVL tree implementation provided in the text (and posted on piazza) is missing two rotation methods:

rotateWithRightChild (AvlNode k2) – a single rotation required because k2 is out of balance due to an insertion of a new node into the right subtree of k2's right child. Note, Weiss did the other single routine for us, **rotateWithLightChild**, which is shown on page 135

doubleWithRightChild (AvlNode k3) a double rotation required when k3 is out of balance due to the insertion of a new node into the left subtree of k3's right child. Note, Weiss did the other double rotation for us, **doubleWithLeftChild** is on page 136.

1. Implement **rotateWithRightChild**, modeling your code after that provided (pages listed above).
2. Implement **doubleWithRightChild**, modeling your code after that provided (pages listed above).
3. Implement a more efficient version of **doubleWithRightChild** without the inefficiency of doing two single rotations – that is, write code that does not call other methods of single-rotation methods.

Make sure that the code that you submit contains both doubleWithRightChild methods, just have one commented out when you submit. For your testing purposes, comment out whichever one you are not testing.

Include a test driver that shows that calls the methods.

What to submit: Submit AvlTree.java to OAKS **no later than Tuesday, August 4, 10:00pm**.

Note: so that you can see what your trees look like, I've included a displayTree method that you can use. Null links are shown as ____. And this method (actually two methods) requires two imports:

```
import java.util.LinkedList;
import java.util.Queue;
```