CPS305 Lab 4

Topics:

- AVL

Files:

- Lab4.py
- Incorrectly named files will receive a mark of zero.

Submit Files:

- Submit file through D2L
- Your file should not have any print statements in it.
- You will lose a mark for each print statement.
- There is a getStudentNumber() function in the Lab4.py file.
- Change the output string to student number.
- You MUST have your student number filled out to receive the mark for the lab.

Lab Description:

In this lab, you are implementing a linked solution of your own AVL. You are given Python class templates of the MyAVL class. Some of the functionalities are given and you are to complete the rest as indicated by the requirements. You will need to copy in some functionality you coded in Lab 3.

Requirements:

- First, create a method called getStudentNumber() that returns your TMU student number. If this does not work, you will receive a mark of zero.
- Next, copy your code MyBST from Lab3.py into the Lab4.py file. You will need working versions of updateHeight(), __contains__(), findSmallest() and findLargest(). If you did not do Lab 3, you will need to implement these methods now.
- A class called MyAVL that extends MyBST is given. You must implement a new remove() method to override the MyBST remove() method. You also must implement the methods reBalance() and getBalanceFactor().
- The getBalanceFactor() method returns the balance factor of the current node.
- The reBalance() method should rebalance the AVL subtree that is out of balance and return the newly balanced subtree. This method will use the functions rightRotate() and leftRotate(), which are given.

Restrictions:

- You cannot use Python's list in your implementation. This includes the methods appends, pop, etc. This lab requires linked implementations of BST and AVL

Testing Your Code:

You can test your code with the tester. The tester is not a be all and end all. You should rely on your own testing before using the tester. Note that the lab should be done on Python 3 and the tester only works on Python 3. The tester is named Lab3Tester.py. Your file must be named correctly and be in the same folder of the tester. Here is an example of executing it on the command line (note that the $ is representative of the beginning of the new line).

```
$ python3 Lab4-Tester.py
```