CS 251, Spring 2009
Voicu Popescu
Assignment 8
Due Friday April 3rd, 11:59PM

AVL Tree

1. Implement a C++ AVL Tree class **AVLTree** in files AVLTree.cpp and AVLTree.h
   a. The items stored are positive integers which serve both as keys and as elements.
   b. The constructor **AVLTree** () builds the AVLTree.
   c. The void **insert**(int *k*) method inserts the key *k* into the AVLTree, and does not do anything if the key is already stored.
   d. The void **remove**(int *k*) method removes key *k* from the AVLTree and does not do anything if AVLTree does not contain key *k*.
   e. The void **printInorder** () method prints the tree to the standard output using an inorder traversal; it prints a (key, height) pair for each node and ends with a newline.
   f. The void **printPostorder** () method prints the tree to the standard output using postorder traversal; it prints a (key, height) pair for each node and ends with a newline.
   g. The destructor **~ AVLTree ()** deallocates all dynamically allocated memory.

   h. Additional instructions:
      i. A test file "AVLTreeTest.cpp" is also provided to test the data structure. Examine and run that program to test your operations. You will not turnin the test file. Your AVLTree class should work with any valid main file. DO NOT include your code in the test file. Implementing AVLTree functionality in the test file instead of the class files will most likely lead to a 0 for the overall assignment grade.
      ii. Example input and output are provided in the files "sampleinput" and "sampleoutput". The provided test file takes keyboard input, and you could use "AVLTreeTest < sampleinput" on lore machine to redirect the sampleinput file as the input. The output should be printed on the screen and you could also use "AVLTreeTest < sampleinput > sampleoutput" to save the output into a file. Please make sure that your program can EXACTLY reproduce that output file given the input file. You may want to use the `diff' program to test this.
      iii. DO NOT change any of the function signatures. Doing so will most likely lead to a compile error when using and a 0 for your overall program grade.
      iv. Be sure to test your program thoroughly.
      v. Please use the Makefile provided.
      vi. Do not partition your code into files we haven't asked for.
      IMPORTANT: Any files other than AVLTree.cpp and AVLTree.h WILL NOT BE GRADED!!!

2. Turn in instructions
   a. Your codes MUST compile on lore.cs.purdue.edu for it to be graded.
   b. All of your documents (AVLTree.cpp and AVLTree.h) MUST be in the directory A8.
   In the directory above A8, run the command on lore.cs.purdue.edu
   ```
   turnin -c cs251 -p A8 A8
   ```

   optionally run the following command to verify what files were submitted
   ```
   turnin -c cs251 -p A8 –v
   ```

Warning: turnin will be automatically disabled at 04/03/2009 at 11:59 PM EST and you will be unable to turnin your assignment after this time.

Warning: turning in multiple times is ok but only the last one will be graded as each turnin permanently overwrites the previous one.

Warning: failure to turnin exactly according to the instructions given above will result in your A8 receiving a grade of 0.