CSCI-1200 Computer Science II — Fall 2008
Lab 11 — Advanced Trees

Checkpoint 1

Download these files:

```
http://www.cs.rpi.edu/academics/courses/fall08/cs2/labs/11_trees/cs2set.h
http://www.cs.rpi.edu/academics/courses/fall08/cs2/labs/11_trees/test_cs2set.cpp
```

Implement and test the decrement operator for `tree_iterator`. Determine the appropriate sequence to *insert* the numbers 1-15 such that the resulting tree is *exactly balanced*. After using the `print_sideways` function to confirm the construction of this tree, test your iterators on the structure. Similarly, create a couple unbalanced trees to demonstrate that both the increment and decrement operators for iterators are debugged. Your decrement operator should correctly decrement the `end()` iterator. You can use the same "trick" we used in Lab 7 to make this work for `cs2list` iterators. The solution code for linked lists is on the webpage. Ask a TA if you have any questions.

To complete this checkpoint: Show one of the TAs your iterator decrement code and your tests cases.

Checkpoint 2

Add a member function called `accumulate` to the public interface of the `cs2set<T>` class, and provide its implementation. The function should take only one argument and it should return of *accumulating* all the data values stored in the tree. The argument is the initial value to start from. The function should only use `operator+=` on type `T`.

Test your code by showing that this works for both a set of ints, where the accumulate function should sum the values in the set, and a set of strings, where the accumulate function should concatenate the strings in the set. Does it matter if the `operator+=` for type `T` is commutative? How can you control the result of accumulate if it is not commutative?

To complete this checkpoint: Show a TA your completed and tested program.