

**Assignment 1**  
Due Feb 2, 2010

Produce code with the following functionality:

1. Implements a binary search tree with addition. You don't have to implement delete on the tree.
2. The nodes in the tree contain strings.
3. Print out the elements in the tree in reverse alphabetic order.
4. Print out the elements in the tree that end in either an 'n' or an 's' in alphabetic order.

You are to write your own binary search tree code. Do not use any existing binary search tree library. You can use non-BST code from your language's api. The binary search tree is to be pointer based, not embedded in an array (or vector).

**Grading**

Item	Percent of Grade
Working Code	30%
Unit Tests	20%
Comments	20%
Quality of Code	30%

**Working Code.** How well your code meets the functional requirements listed above accounts for 30% of the grade for the assignment.

**Unit Tests.** Having unit tests that cover the functionality of your code accounts for 20% of your grade. The unit tests are to be scalable. Tests using just print statements are not scalable.

See <http://www.eli.sdsu.edu/courses/fall03/cs535/notes/testing/testing.html> for notes on unit testing (SUnit) Smalltalk. See <http://www.eli.sdsu.edu/courses/spring09/cs580/notes/SourceControlTesting.pdf> for notes on unit testing (JUnit) in Java. See <http://www.junit.org/> for more information about JUnit.

For C++ and C# you can get the equivalent testing framework at <http://www.xprogramming.com/software.htm>

**Comments.** Having the appropriate comments in your code will count for 20% of the grade.

**Quality of Code.** Having good quality of code counts for 30% of the grade. Quality of code includes formatting, names and modularity.

For more information about comments and quality of code see the lecture notes of past CS535 courses. For even more information see Code Complete 2, McConnell, Microsoft Press, 2004. Smalltalk programmers can look at Smalltalk Best Practice Patterns, Beck, Prentice Hall PTR, 1997

### **What to Turn in**

Turn in hard copy of your code and unit tests. On your assignment indicate if you are currently enrolled in the class or are trying to crash the class.

### **Late Policy**

An assignment turned in 1-7 days late, will lose 3% of the total value of the assignment per day late. The eight day late the penalty will be 40% of the assignment, the ninth day late the penalty will be 60%, after the ninth day late the penalty will be 90%. Once a solution to an assignment has been posted or discussed in class, the assignment will no longer be accepted. Late penalties are always rounded up to the next integer value.