**COMP1917 Computing 1  
Session 2, 2014**

**Week 11 Tutorial - Binary Search Trees**

**This page was last updated: 10/11/2015 18:30:22**

**Presentation Topic for Week 12**

Any topic that was missed in earlier weeks, or a repeat of any topic that was previously covered, or any new topic of your own choosing (relevant to the course).

1. Draw the Binary Search Tree that would result from inserting these numbers, in order, into an initailly empty tree:

5, 8, 1, 3, 9, 6, 4, 7, 2

1. Write a function
2. int sumTree( Tnode \*root )

which, given the root node of a Binary Tree, returns the sum of the data components for all nodes in the tree.

1. Write a function list\_from\_tree() which, given a Binary Search Tree, constructs a Linked List containing the same data items as the tree, in ascending order. The prototype for the function should be:
2. Lnode \* list\_from\_tree( Tnode \*root, Lnode \*list );

At the top level, it would be called like this:

head = list\_from\_tree( root, NULL );

Hint: traverse the tree recursively and build the list from back to front, starting with the largest item and working backwards to the smallest item.

1. Bonus Challenge (if time permits). Write a function
2. Tnode \*remove\_node( Tnode \*root, Tnode \*node )

which, given pointers to the root of a Binary Search Tree and a node within that tree, removes the specified node while preserving the BST property, and returns a pointer to the root of the resulting tree.   
Hint: Try to use recursion, and consider that the node may have no children, one child, or two children; if it has two children, try replacing the node with the next (in-order) node in the tree.

1. Any Questions left over from previous weeks, or about Assignment 2, or any part of the course.