COMP 210 **Object-Oriented Programming**

Lab 5 & Homework 5

February 14 *Spring 2018*

Abstract

This week we'll work out some of the implementation for a basic node-based Binary Search Tree.

1 Lab 5

Design a binary search tree class (or class hierarchy) containing primitive integer-type data. You are free to design this as you wish (including using the simple version we worked through in class). Your class(es) should contain:

- Private constructor(s)
- A static factory method for constructing a search tree from an array of unsorted integer. The factory will construct an ideally balanced initial tree given the number.
- An `isEmpty` predicate
- A predicate to check if a specific value is contained in the tree
- Mutators for insertion and deletion, both of which should return `true` if they were successful and `false` otherwise
- A mutator that will balance the tree.

Use Eclipse to auto-generate code wherever possible. Once the design is setup (documentation, declarations, stubs, tests), turn your attention to the implementation. Even though you may reuse code discussed in class, much of that code could be refactored into something better. Go through the entire top-down method design process again:

1. Analyze the problem and decide if any helpers could be used to make the implementation simpler.
2. Declare and stub each helper in your class(es).
3. Complete the implementation of the method using the helpers.
4. Repeat the process for the helpers

Submit as assignment `hwk5`; due by the start of lab on **Wednesday 2/21**.