CS 474: Object Oriented Programming Languages and Environments

Spring 2014

Objective C project (optional)

Due time: 7:00 pm on Saturday 4/5/2014

You are to implement a *set calculator app* in the Objective C language with Apple's iOS as a target platform. The calculator allows an interactive user to enter and edit information about two sets containing integer numbers. As with the first project, you must implement each set as a Binary Search Tree (BST). However, the requirements on this project are somewhat simplified with respect to first project, as you will see below. At any point in time, the user will edit and modify one the two BSTs, that is, S_1. When the editing is complete, the user can save S_1 as the second BST, S_2. This operation will deep copy the first set (S_1) into the second set; however, S_1 is not modified. This set could be edited further, independently of S_2. Set operations take place between S_1 and S_2; operation results are always stored in S_1.

Recall that BSTs are binary trees subject to the following properties. First, a numeric value is associated with each node. Second, each node can have at most two children, a left child and a right child. Third, given a node *x*, the values of all nodes in the left subtree of *x* are less than the value of *x*, and the values of all nodes in the right subtree of *x* are greater than the value of *x*. No duplicate values will be allowed in your trees.

The GUI of your program should support the following functionality; you should choose an appropriate XCode widget to implement each piece of functionality.

1. *Clear set.* This function allows interactive users to delete the current S_1 set. The previous value stored in S_1 is lost.

2. *Switch sets.* The sets associated with S_1 and S_2 are swapped, meaning that S_1 will receive the previous S_2 set and vice versa.

3. *Save set.* The S_1 is deep copied into S_2. The previous content of S_2 is lost. The content of S_1 is not affected. The two sets must not share any data structures, that is, they can be modified independently of each other.

4. *Display set contents.* The numeric values stored in the two sets are displayed as two ordered lists (non-decreasing order) in an appropriate widget or widgets. The two sets are not modified.

5. *Add element.* This function allows a user to add a new integer to S_1. The value is read from an appropriate *line input* widget. No action is taken if the number in question is already in the set. The insertion should preserve the BST properties of S_1.

6. *Union.* This element takes the set union of S_1 and stores the resulting value in S_1. The previous content of S_1 is lost. S_2 is not modified by this operation.

You must work alone on this project. **You are required to implement all set operations on the binary search tree representation of sets. You are not allowed to use alternative representations of sets, such as predefined Cocoa Touch classes.** You are not allowed to discuss designs or share code with other students. However, you are encouraged to use the Blackboard discussion board to post or answer questions about specific aspects of the project.

To implement this project, you will be required to use an Apple Mac computer with the XCode developer environment, version 4.3 or above. Note that the Automatic Reference Counting (ARC) feature was introduced in version 4.3. If you do not own a Mac, you may use the ICL of the Computer Science

Department or you may rent time from the service http://www.macincloud.com. An educational discount allows five (eight) hours of daily use for $20 ($30), according to the their web site.

To turn in your project, please submit a .zip archive containing your entire XCode project and a text file named README.txt discussing how to use your iOS app.

Good luck!