CS 474: Object Oriented Programming Languages and Environments

Fall 2014

Objective C project

Due time: 9:00 pm on Monday 11/17/2014

This project is similar to the first Smalltalk project in that you are to implement a *set calculator app*, this time in the Objective C language with Apple's iOS as a target platform. As with the first project, you will define a SetManager superclass with two concrete subclasses for managing sets using different data structures. In particular, subclass ListSetManager will implement sets as linked lists whereas subclass OCSetManager will use the predefined Objective C class NSMutableArray for sets. In contrast to Smalltalk, Objective C does not define a linked list class. Thus, you will have to define your own class from scratch.

As with the Smalltalk project, a set manager must work with two sets, which we will call *A* and *B* here. For simplicity the sets contain integer numbers. No duplicate values will be allowed in each set, regardless of the chosen set implementation.

The GUI of your program should support the following operations; you should choose an appropriate XCode widget to implement each piece of functionality. The set manager defines storage and operations for sets *A* and *B*. Some of the operations must be implemented in superclass *SetManager* as opposed to the subclasses. When this is not indicated, you are free to implement operations in subclasses ListSetManager and OCSetManager.

- Start list manager — The current instances of sets *A* and *B* are discarded. This widget creates two instances of class ListSetManager to be the new *A* and *B* sets. Both sets will be initially empty.
- Start OC Manager — The current instances of sets *A* and *B* are discarded. This widget creates two instances of class OCSetManager to be the new *A* and *B* sets. Both sets will be initially empty.
- Clear — This operation resets set *A* to be the empty set. No new set instances are created.
- Switch — This operation swaps the sets associated with *A* and *B*, meaning that *A* will receive the previous *B* set and vice versa. This operation must be implemented in superclass SetManager.
- Save — This operation copies the *A* set into *B*. The previous content of *B* is lost. The content of *A* is not affected.
- Add element — This operation allows a user to add a new number to *A*. No action is taken if the number in question is already in the set. Otherwise, the number is added to the set.
- Remove element — This operation allows a user to remove a number from *A*. No action is taken if the number in question is not in the set. Otherwise, the number passed as an argument is removed from *A*.
- Size — This operation displays the number of elements currently stored in set *A*.
- Indexed access — This operation takes as input an integer *i* and returns the element at position *i* in set *A*.
- Membership test — This operation returns true or false depending on whether set *A* contains the argument number. This operation must be implemented in superclass SetManager.

- Union — This operation computes that set union of sets A and B. The result is stored as set A. The previous content of A is lost. B is not modified by this operation. This operation must be implemented in superclass SetManager.

You must work alone on this project. You are not allowed to discuss designs or share code with other students. However, you are encouraged to use the Piazza 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!