CS 635 Advanced OO Design and Programming
Spring Semester, 2011
Assignment 3

San Diego State University -- This page last updated 3/3/11

Assignment 3
Max-Heap

Due Mar 8 23:59

1. Review your units tests for adding elements to the Max Heap from assignment 1. Make sure that the tests adequately test adding elements to the heap and obtaining the elements in the heap that are odd. Record those tests. When you are done with the assignment determine how good the tests were. That after making the changes required in this assignment you were confident that worked after running the tests. Did you have add to or modify your tests?

2. Refactor your tree code to use standard names for methods, remove helper methods on the heap that deal with heap nodes, and any other clean up you feel is needed in your code. You might find the refactorings rename and move useful here. In Eclipse these refactorings can be found in the Refactoring menu.

3. The heap class in assignment two is a collection. Determine the correct location in your language's collection class hierarchy. Find **all** methods that you need to implement in-order to add your class in the language's collection class hierarchy. (C++ people get a pass on this problem as STL is painful to subclass.).

4. Make the parent class of your binary search tree the parent determined in problem 1. Rename your existing methods to conform to the collection classes standards. One may need to stub some methods to satisfy the parent class's constraints. Note we will only be interested in implementing a few of these methods. You do not have to implement all the methods in the parent class. We will need at least the add method, toArray method and toString method.

5. Implement what we will for now will call OddIterator. Using Java syntax the class will have the methods given below. C#, Ruby and C++ people may need to implement different methods to conform to their language's conventions.

   OddIterator(Array<int> input) - constructor

   boolean hasNext() - returns true if the iteration has more elements are odd.

   next() - returns the next element in the iteration that is odd.

6. Create a decorator class OddHeapDecorator that decorates your new Heap class. The decorator modfies the toString method and toArray elements to return just odd elements. Does this make sense as a decorator?

Grading

Item	Percent of Grade
Working Code	15%
Unit Tests	10%
Proper implementation of Patterns	60%
Quality of Code	15%

Turning in your Assignment

For the people not using Smalltalk. To turn in your assignment combine all your files for the assignment into a zip file. (No rar files) and upload them in the course portal (Under the assignments tag click on assignment 3).