

Assignment 16---100 points
(Java Programming Practices)

(Bonus Deadline: 15% bonus for submission by Thursday, 12/15/2011, Midnight at Moodle)
(Submission by 12/19/2011/Monday-Midnight for full credits)
(No Late Submissions Accepted)

Your name:	Score:
------------	--------

Complete the following Java programming projects:

19.5 (Bubble Sort) Implement bubble sort—another simple yet inefficient sorting technique. It is called bubble sort or sinking sort because smaller values gradually “bubble” their way to the top of the array (i.e., toward the first element) like air bubbles rising in water, while the larger values sink to the bottom (end) of the array. The technique uses nested loops to make several passes through the array. Each pass compares successive pairs of elements. If a pair is in increasing order (or the values are equal), the bubble sort leaves the values as they are. If a pair is in decreasing order, the bubble sort swaps their values in the array.

- The first pass compares the first two elements of the array and swaps their values if necessary. It then compares the second and third elements in the array. The end of this pass compares the last two elements in the array and swaps them if necessary. After one pass, the largest element will be in the last index. After two passes, the largest two elements will be in the last two indices. Explain why bubble sort is an $O(n^2)$ algorithm.

19.8 (Recursive Linear Search) Modify Fig. 19.2 to use recursive method *recursiveLinearSearch* to perform a linear search of the array. The method should receive the search key and starting index as arguments. If the search key is found, return its index in the array; otherwise, return -1 . Each call to the recursive method should check one index in the array.

19.9 (Bonus 40 points) (Recursive Binary Search) Modify Fig. 19.4 to use recursive method *recursiveBinarySearch* to perform a binary search of the array. The method should receive the search key, starting index and ending index as arguments. If the search key is found, return its index in the array. If the search key is not found, return -1 .