Name: ***Jonathan Smoley***

Assignment: 1

Complete & Correct:	42 / 42
Tests:	6 / 6
Format and Comments:	6 / 6
Writeup:	6 / 6
Total Score:	60 / 60

Comments:

1. In regards to your question about access to local files on ada, you can "upload" files to ada via VS Code. The easiest way to do this is just drag and drop a local file in VS Code (in one window) to your folder on ada (in another window). You don't need any special extensions for dragging and dropping like this.

Unit test output:

```
[==========] Running 15 tests from 3 test suites.
[----------] Global test environment set-up.
[----------] 5 tests from BasicBubbleSortTest
[ RUN      ] BasicBubbleSortTest.EmptyArray
[       OK ] BasicBubbleSortTest.EmptyArray (0 ms)
[ RUN      ] BasicBubbleSortTest.ReverseOrder
[       OK ] BasicBubbleSortTest.ReverseOrder (0 ms)
[ RUN      ] BasicBubbleSortTest.InOrder
[       OK ] BasicBubbleSortTest.InOrder (0 ms)
[ RUN      ] BasicBubbleSortTest.PartiallyUnorderedList
[       OK ] BasicBubbleSortTest.PartiallyUnorderedList (0 ms)
[ RUN      ] BasicBubbleSortTest.UnorderedList
[       OK ] BasicBubbleSortTest.UnorderedList (0 ms)
[----------] 5 tests from BasicBubbleSortTest (0 ms total)

[----------] 5 tests from BasicInsertionSortTest
[ RUN      ] BasicInsertionSortTest.EmptyArray
[       OK ] BasicInsertionSortTest.EmptyArray (0 ms)
[ RUN      ] BasicInsertionSortTest.ReverseOrder
[       OK ] BasicInsertionSortTest.ReverseOrder (0 ms)
[ RUN      ] BasicInsertionSortTest.InOrder
[       OK ] BasicInsertionSortTest.InOrder (0 ms)
[ RUN      ] BasicInsertionSortTest.PartiallyUnorderedList
[       OK ] BasicInsertionSortTest.PartiallyUnorderedList (0 ms)
[ RUN      ] BasicInsertionSortTest.UnorderedList
[       OK ] BasicInsertionSortTest.UnorderedList (0 ms)
[----------] 5 tests from BasicInsertionSortTest (0 ms total)

[----------] 5 tests from BasicSelectionSortTest
[ RUN      ] BasicSelectionSortTest.EmptyArray
[       OK ] BasicSelectionSortTest.EmptyArray (0 ms)
[ RUN      ] BasicSelectionSortTest.ReverseOrder
[       OK ] BasicSelectionSortTest.ReverseOrder (0 ms)
[ RUN      ] BasicSelectionSortTest.InOrder
[       OK ] BasicSelectionSortTest.InOrder (0 ms)
```

```
[ RUN      ] BasicSelectionSortTest.PartiallyUnorderedList
[       OK ] BasicSelectionSortTest.PartiallyUnorderedList (0 ms)
[ RUN      ] BasicSelectionSortTest.UnorderedList
[       OK ] BasicSelectionSortTest.UnorderedList (0 ms)
[----------] 5 tests from BasicSelectionSortTest (0 ms total)

[----------] Global test environment tear-down
[==========] 15 tests from 3 test suites ran. (0 ms total)
[  PASSED  ] 15 tests.
```

Performance test output:

```
# All times in milliseconds (msec)
# Column 1 = input data size
# Column 2 = avg time bubble sort, sorted
# Column 3 = avg time bubble sort, reversed
# Column 4 = avg time bubble sort, shuffled
# Column 5 = avg time insertion sort, sorted
# Column 6 = avg time insertion sort, reversed
# Column 7 = avg time insertion sort, shuffled
# Column 8 = avg time selection sort, sorted
# Column 9 = avg time selection sort, reversed
# Column 10 = avg time selection sort, shuffled
0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
1000 0.00 4.33 3.33 0.00 2.00 1.00 2.00 2.00 1.67
2000 0.00 9.33 7.00 0.00 5.00 2.00 5.00 4.00 5.00
3000 0.00 20.00 16.00 0.00 12.00 6.00 12.00 10.00 12.00
4000 0.00 36.00 33.00 0.00 21.33 10.00 21.00 19.00 22.00
5000 0.00 57.33 46.00 0.00 34.00 16.00 33.00 30.00 34.00
6000 0.00 145.33 109.33 0.00 110.00 53.00 48.00 44.00 49.00
7000 0.00 176.67 138.33 0.00 150.00 73.00 66.00 59.00 67.00
8000 0.00 247.67 230.00 0.00 196.00 95.00 86.00 78.00 87.00
9000 0.00 285.33 225.00 0.00 249.00 120.00 109.00 98.00 110.00
10000 0.00 367.33 301.00 0.00 307.00 149.00 135.00 122.00 136.00
```