LESSON 39 NOTES

The Insertion Sort Algorithm

Precondition: ***Values in array are primitive data types.***

1. ***Begin with a list containing ONE value only (consider it to be sorted)***
2. ***"Pick up" a new value (the second value in the list)***
    i. ***Sort the first two values (only swap if necessary)***
3. ***Repeat the above with sequential elements in the list***

An Example:

```
3 is sorted.
Shift nothing.                :     3     9     6     1     2
Insert 9.

3 and 9 are sorted.
Shift 9 to the right.         :     3     9 ──► 6     1     2
Insert 6.

3, 6 and 9 are sorted.
Shift 3,6,& 9 to the right.   :     3 ──► 6 ──► 9 ──► 1     2
Insert 1.

1, 3, 6 and 9 are sorted.
Shift 9,6,& 3 to the right.   :     1     3 ──► 6 ──► 9 ──► 2
Insert 2.


All values are sorted.        :     1     2     3     6     9
```

The Bubble Sort Algorithm

Precondition: *Values in array are primitives*

1. *Begin first "pass" (sequence) through a list*
2. *Compare adjacent items in the list*
3. *If two items are out of order (element1 > element2), swap them*
4. *Repeat steps #1 to #3 until no exchange is made on a "pass"*

Exchange (Bubble) Sort

Element	1	2	3	4	5	6	7	8
Data	27	63	1	72	64	58	14	9
1st pass	27	1	63	64	58	14	9	72
2nd pass	1	27	63	58	14	9	64	72
3rd pass	1	27	58	14	9	63	64	72...

The first two data items (27 and 63) are compared and the smaller one placed on the left hand side. The second and third items (63 and 1) are then compared and the smaller one placed on the left and so on. After all the data has been passed through once, the largest data item (72) will have "bubbled" through to the end of the list. At the end of the second pass, the second largest data item (64) will be in the second last position. For n data items, the process continues for n-1 passes, or until no exchanges are made in a single pass.