Selection Sort

Putting data in order

int []x= {7,1,9,3,5,4}

| 7 | 1 | 9 | 3 | 5 | 4 |

- Sort the array in ascending order:

- Remember the bottom of this list is index 0 and the top of this list is index 5.

int []x= {7,1,9,3,5,4}

| 7 | 1 | 9 | 3 | 5 | 4 |

- Begin the sort by looping through the array and locating the largest value.

- Place the largest value at the top of the list by swapping the values.

| 7 | 1 | 4 | 3 | 5 | 9 |

- Continue the sort by looping through the array up to the index 4 and locating the largest value.

- Place the largest value at index 4 by swapping the values.

5	1	4	3	7	9

- Continue the sort by looping through the array up to the index 3 and locating the largest value.

- Place the largest value at index 3 by swapping the values.

3	1	4	5	7	9

- Continue the sort by looping through the array up to the index 3 and locating the largest value.

- Place the largest value at index 3 by swapping the values.

3	1	4	5	7	9

- Continue the sort by looping through the array up to the index 2 and locating the largest value.

- Place the largest value at index 2 by swapping the values.

| 3 | 1 | 4 | 5 | 7 | 9 |

- Continue the sort by looping through the array up to the index 1 and locating the largest value.

- Place the largest value at index 1 by swapping the values.

1	3	4	5	7	9

- Continue the sort by looping through the array up to the index 1 and locating the largest value.

- Place the largest value at index 1 by swapping the values.

The method selectSort uses a selection sort to arrange an array of **double** values in ascending order.

```java
public static void select Sort (double[] list)
{
    for (int top = list.length - 1; top> 0; top--)
    {
        int largeLoc = 0;
        for (int i = 1; i <= top; i++)
            if (list[i] > list[largeLoc])
                largeLoc = ij

        double temp = list [top] ;
        list[top] = list [largeLoc] ;
        list[largeLoc] = temp;
    }
}
```