Opening Discussion

- Do you have any questions about the IcP?
- Minute essay comments:
  - How to read the Scala API?
  - What would happen with an infinite loop that opens windows?
  - Move with click? Click and drag?
  - Changing fonts.
  - How do I feel about Spurs-Timmy?
There are many reasons that you might want the data you are working with to be in a particular order.

If nothing else, humans often like seeing things in certain orders.

It turns out that ordered data can be beneficial for the computer as well.

Putting things in order by some value is called sorting.
Methods of Sorting

- If I ask you to sort a bunch of items, how would you go about doing it? Describe your approach.
- How does it vary for different types or configurations of objects?
We are going to look at three different sorting techniques today.

These sorts all do work that is proportional to the square of the number of elements.

That isn't good for large collections, but the sorts are fairly simple to write.

These work “in place” so we use arrays.

Each involves an inner loop that reorders things and an outer loop that makes the inner one happen over and over.
Bubble Sort

- **Inner loop:**
  - Compare adjacent elements and swap them if they are out of order.

- **Outer loop:**
  - Repeat n-1 times or until no swaps are done.
  - The latter option is called a flagged bubble sort.
Selection Sort

- This is often called a min-sort or a max-sort depending on how you write it. I'll describe a min-sort here.
- Inner loop:
  - Find the smallest element and SWAP it into position if not already there.
- Outer loop:
  - Repeat n-1 times so all elements are in the right place.
- Does only $O(n)$ swaps, but still $O(n^2)$ comparisons.
Insertion Sort

- **Inner loop:**
  - Take the next element and shift it down to the right spot.

- **Outer loop:**
  - Run through all the elements starting with the second.

- This sort is actually a bit faster (factor of 2) on random data. It is really efficient on nearly sorted data.
Watching Them Work

- One advantage of doing graphics before sorting is that we can write code to visualize what is happening when we sort numbers with these sorts.
- Let's write this code and watch our sorts work.
Show me what would happen after each iteration of the inner loop if we min-sort these values.

- 4, 7, 1, 3, 8, 2

The IcP is on Monday.