Object-Oriented Programming unit:

- Review of object-oriented concepts (**Today** and next week Monday)
- Review of linked lists (next week Wednesday)
- Version control systems (next week Friday)
- Documentation; Java GUI components (week-after Monday)
- Abstract data types and Java Collections (week-after Wednesday and Friday)

Today:

- Go over practice problems
- Lab retrospective
- Defining terms carefully
- Writing a class
- Static stuff
- Polymorphism

```c
void bubbleSort(int array[], int n)
{
  int i, j;

  for (i = n; i > 1; i--)
    {
      for (j = 1; j < i; j++)
        {
          if (array[j] < array[j-1])
            {
              int temp = array[j];
              array[j] = array[j-1];
              array[j-1] = temp;
            }
        }
    }
}
```

```
double findSum(double[] array, int period, int n)
{

        double sum = 0;

        for (int i = 0; i < period; i++)

             for (int j = i; j <n; j += period)

                       sum += array[j];
      return sum;
}
```

Lesson 1: *Use objects to encapsulate data and functionality. Use classes to define a new type of object.*

Lesson 2: *The rest of the program needs to know an object's interface. Effectively, the object's interface is its type. The* `interface` *construct allows us to make distinct classes to be subtypes of the same supertype.*

Coming up:

- *Read/review Savitch Chapter 4 and Sections 5.1, 8.1, and 13.1, as necessary.*
- **Due Mon, Feb 2.** *Do Project 1, "Quick sort and experimental comparisons."*
- **Due Thurs, Feb 5.** *Read pre-lab reading 4, take quiz. (Coming soon. . .)*
- **Due Wed, Feb 18.** *Do Project 2, "First Calculator."*