Program 1 Questions?

Any last-minute program 1 questions before we get started?

It's due at 10pm tonight!!

Arrays Review

What does this code do:

```
int[] arr = {1,2,3,4,5,6,7,8,9,10};
for (int i=0; i<arr.length/2; i++) {
    int temp = arr[i];
    arr[i] = arr[arr.length-(i+1)];
    arr[arr.length-(i+1)] = temp;
}
```

Introducing: Methods

We've seen methods before:

1. `char z = str.charAt(0);`
2. `int sq = Math.pow(x,2);`
3. `int y = rand.nextInt(5);`

They've basically functioned as black boxes - we put parameters in, and get something out.

Why methods?

A method is just a bunch of instructions with a name - our `main` methods are methods.

We used arrays and a `for` loop to calculate an average last time. We could write this code over and over every time we wanted to calculate an average, or we could put it in its own method.

Why methods?

Methods help make code easier to understand and reduce the chance of making a stupid mistake:

1. Increase modularity
2. Increase readability/maintainability
3. Reduce redundancy

Using methods

Up to this point, when we've used other methods, all we've needed to know is:

1. What types of variables to put in and how many
2. What type of result we get

...along with, of course, a vague idea of what the method does (e.g. "calculate average"). But when you're using a method, you don't need to know exactly what the instructions inside the method are.

`static` methods

We'll look first at creating `static` methods. We've used these; they have the form:

```
<Class>.<method>(<params>);
```

Compare these to object methods, which we've also used, which have the form

```
<variable id>.<method>(<params>);
```

Examples of both kinds?

Writing `static` methods

Let's write a method to compute the average of an array of numbers. (`Math` doesn't have one!)

We'll need to:

1. Pick a name for the method
2. Figure out what the parameters are
3. Figure out what our result will be
4. Add `public static` modifiers