Last Time

• Iterative vs. Recursive Code
  – Iterative and Recursive Factorial
  – Recursive Fibonacci
New Programming Challenge

- What is Fibonacci number 73?
- How about Fibonacci number 42?
- Fibonacci number 29?

- What is annoying about asking for them in this order?
  - We need to calculate 29 and 42 to get to 73, but we don’t store those values and need to calculate them a second time (and a third time for 29).
Saving Fibonacci Values

```java
long fib1 = 0;
long fib2 = 1;
long fib29, fib42;
int goal = 73;

for (int i = 2; i <= goal; i++) {
    long temp = fib1 + fib2;
    if (i == 42) {
        fib42 = temp;
    } else if (i == 29) {
        fib29 = temp;
    } //if-else
    fib1 = fib2;
    fib2 = temp;
} //for
```
Saving Fibonacci Values

```
int a=0, b=1, c, d, e, f, g, h, i, j;
c = a + b;
d = b + c;
e = c + d;
f = d + e;
g = e + f;
h = f + g;
i = g + h;
...```

Better Solution: Arrays

• Consecutive blocks of data in memory

```java
int fibo[] = new int[100];
fibo[0] = 0;
fibo[1] = 1;
```
Array Properties

• We can put any type of object in an array: integers, doubles, booleans, Strings, Clocks, ...

• We can get the size of the array by calling a.length;

• We can iterate across an array using a for loop!
Calculating Fibonacci Numbers

```java
int fibo = new int[100];
fibo[0] = 0;
fibo[1] = 1;
for (int i = 2; i < 100; i++) {
    fibo[i] = fibo[i-1] + fibo[i-2];
} //for
System.out.println(fibo[72]);
System.out.println(fibo[41]);
```
New Challenge

• Given this array, print the array indices where the smallest and largest values are located, and calculate the average:

  – double[] a = {3, 6.5, 2, -6, 19, 0, 3};

  – min = a[3]
  – max = a[4]
  – avg = 3.92857...
Any Questions?