

The background of the slide is a grayscale image of a circuit board. It features a complex network of black lines representing traces and several solid black circles representing vias or components. The circuitry is arranged in a somewhat symmetrical, horizontal pattern. A solid black horizontal band runs across the middle of the image, partially obscuring the circuit board design. In the center of this dark band, the text 'CMSC 131' is displayed in a large, white, sans-serif font. Below it, the text 'Fall 2018' is written in a smaller, green, monospaced font.

# CMSC 131

Fall 2018

# Announcements

- Project #4 due Sunday

# Processing Arrays

Arrays have a length field:

```
arr.length
```

Standard idiom for processing array:

```
for (int i = 0; i < arr.length; i++) {  
    process a[i]  
}
```

# Examples

ArrayExample0.java

ArrayGame.java

# More memory diagrams with Arrays

Example:

```
int[] x = new int[5];  
x[2] = 7;  
x = new int[5];
```

# Copying Arrays

Does this make a copy?

```
int[] a = new int[1000];
```

...

```
int[] b = a;
```

How can we write code that copies an array?

1. Let's write the code
2. Other ways (we won't cover them):
  - `clone`
  - `System.arraycopy`
  - `Arrays.copyOf`

# Resizing an Array

Once array is instantiated, can you change it's size?

What can we do? Fake it.

Let's code this up:

Suppose we have an array, arr.

- Create a temporary array that is bigger than arr
- Copy existing data from arr over to new array
- Add additional data
- Re-assign arr so that it refers to new array

# Alternate Style

Usually preferred:

```
int[] x, y;  
int a, b;
```

This also works:

```
int x[], y[], a, b;
```

But why would you ever use this?



# Arrays of References

Suppose I have a class called Cat and I want to store a sequence of Cats.

Let's draw the memory diagram for this:

```
Cat[] x;  
x = new Cat[4];    // how many Cats have I made?  
x[0] = new Cat("Fluffy");  
x[1] = new Cat("Princess");  
x[2] = new Cat("Spot");  
x[3] = new Cat("Steve");
```

# Crazy Example

Creating something complex from something simple.

Example: `Word.Java`, `Sentence.Java`,  
`Paragraph.java`, `Driver.java`

## Initializing an array When Constructed

```
char[] arr = {'x', '@', 'A', '!'};
```

```
double[] values = {3.1, 62.79, 5.88, 6.1, 7.55};
```

```
Cat[] kitties = {new Cat("Felix"), new Cat("Tom"),  
                 new Cat("Sylvia"), new Cat("Oscar")};
```

# Mutability

What does it mean for a class to be **mutable**?

**Immutable**?

Can we look at a class and tell?

Always document whether your class is mutable or immutable!

Why is immutable “preferred”?