Review

- Objects
  - data fields
  - constructors
  - Methods
- Classes

Arrays

- A special kind of variable that holds not one, by many data items of a given type.
- Declared like variables, only type is followed by a pair of brackets.

```
float[] xs;
```

- Can be initialized using a special syntax involving the `new` keyword, the type, and a *size* in brackets.

```
int[] diameters = new int[10]; // Ten diameters
```

Arrays

- Individual data items are accessed with an index and square brackets.
  - `diameters[0]`,`diameters[1]`,etc
  - **Indexes start at 0!**
- The length of an array can be determined using its `length` property.
  - `diameters.length`
  - The length of an array is one greater than the last valid index.
- Arrays can be passed to, and returned from functions.

```
int[] diameters = new int[10];

void setup() {
  size(500, 500);
  background(200);

  for (int i=0; i<diameters.length; i++) {
    diameters[i] =  int(random(0, width/2));
  }

  fill(255, 0, 0);
  for (int i=0; i<diameters.length; i++) {
    ellipse(random(width), random(height), diameters[i],
diameters[i]);
  }
}

void draw() {
}
```

Use the Ball class

Treat in a manner very similar to a primitive data type.

```
Ball[] balls = new Ball[20];

void setup() {
  size(500, 500);
  fill(255, 0, 0);
  smooth();
  ellipseMode(CENTER);

  // Create all new Ball objects
  for (int i = 0; i < balls.length; i++) {
    balls[i] = new Ball();
  }
}

void draw() {
  background(255);

  for (int i = 0; i < balls.length; i++) {
    balls[i].update();
    balls[i].draw();
  }
}
```

← Declare an array of Balls.

← New objects are created with the ***new*** keyword.

← Methods of objects stored in the array are accessed using dot-notation.

Built-in Array Functions

append(*array, item*)
- returns a new array expanded by one and add item to end

expand(*array, newSize*)
- returns a new array with size increased to newSize

shorten(*array*)
- returns a new array shortened by one

concat(*array1, array2*)
- returns a new array that is the concatenation of array1 and array2

subset(*array, offset [, length]*)
- returns a subset of array starting at offset and proceeding for length (or end)

splice(*array, value|array2, index*) or
- returns a new array with value or array2 inserted at index

sort(*array*)
- returns a new array sorted numerically or alphabetically

reverse(*array*)
- returns a new array with all elements reversed in order

Pop

- A game that measures your balloon-popping skill.
- How it should work…
  - As game runs, randomly placed balloons inflate
  - When the player pops (clicks on) a balloon, 1 point is earned
  - Points are added up throughout the game duration
  - If one click is over top multiple balloons, all balloons pop and multiple points are earned
  - The game runs for 30 seconds, and then ends