

Arrays Lists and Classes

# VA345 Creative Coding

Course Instructor : Assist. Prof. Dr. Selcuk ARTUT

## Food for thought : Karsten Schmidt

<http://toxi.co.uk/>, <http://postspectacular.com/>



Karsten Schmidt is an award-winning London based computational designer and researcher merging code, design, art & craft skills. Originally from East Germany and starting in the deep end of the early 8-bit demo scene, for the past 25+ years he's been adopting a completely trans-disciplinary way of working and has been laterally involved in a wide range of creative disciplines. With his practice PostSpectacular, he's been actively exploring current possibilities at the intersection of design, art, software development and education and applying this mixture hands-on to a variety of fields, from architecture, branding, digital fabrication, interactive installations, motion graphics & music.

He is a prolific contributor to several large open source projects, was an early contributor to the Processing.org project and to several books about programming and graphic design. His work has been featured and exhibited internationally, including the MoMA New York, Design Museum, Barbican Centre and Victoria and Albert Museum, London.

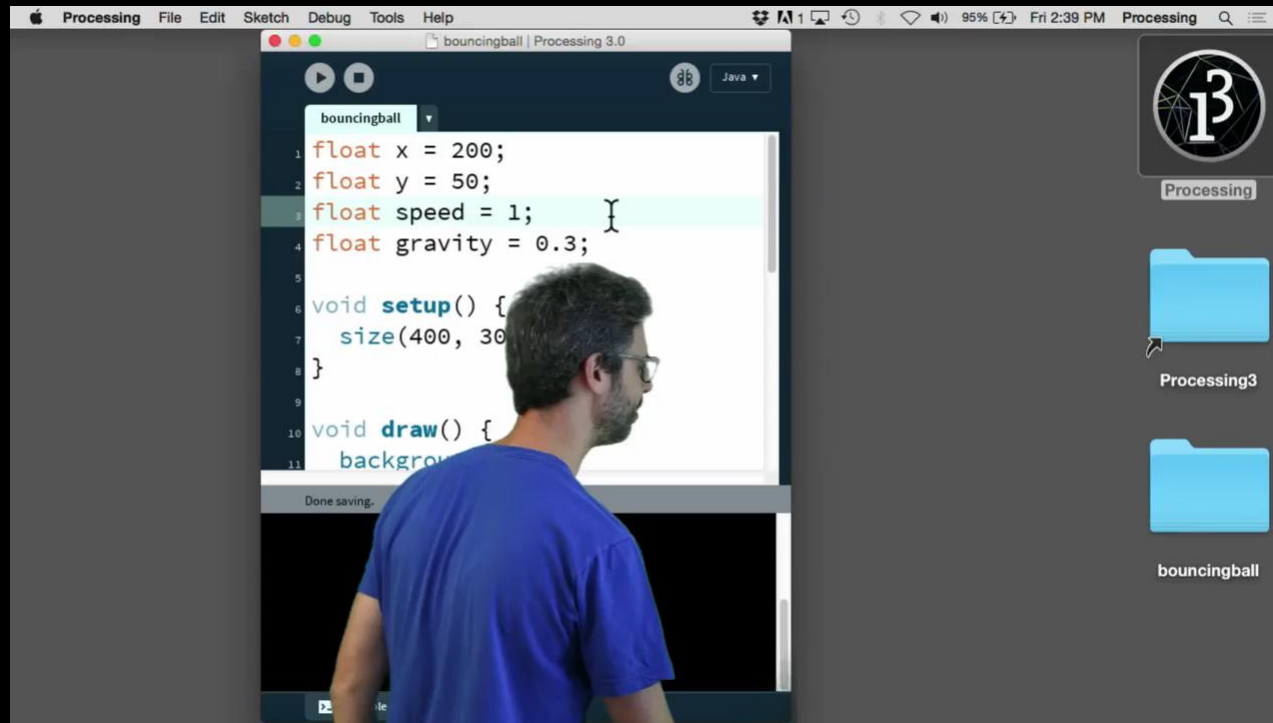
JUXT





# Debugging

Debugging is the process of finding and resolving of defects or problem within the program that prevent correct operation of computer software or a system. (Wikipedia)



Color Theory Recap :

<https://www.youtube.com/watch?v=Qj1FK8n7WgY>

# Arrays

The term array refers to a structured grouping or an imposing number—“The dinner buffet offers an array of choices,” “The city of Los Angeles faces an array of problems.” In computer programming, an array is a set of data elements stored under the same name.

Arrays can be created to hold any type of data, and each element can be individually assigned and read. There can be arrays of numbers, characters, sentences, boolean values, etc. Arrays might store vertex data for complex shapes, recent keystrokes from the keyboard, or data read from a file.

example demonstration

```
int[] x = { 50, 61, 83, 69 }
```

# Arrays

```
int[] x = { 50, 61, 83, 69, 71, 50, 29, 31, 17, 39 };  
int[] y = { 18, 37, 43, 60, 82, 73, 82, 60, 43, 37 };  
beginShape();  
// Reads one array element every time through the for()  
for (int i = 0; i < x.length; i++) {  
  vertex(x[i], y[i]);  
}  
endShape(CLOSE);
```

# Declaring Arrays

```
int[] data; // Declare  
void setup() {  
  size(100, 100);  
  data = new int[5]; // Create  
  data[0] = 19; // Assign  
  data[1] = 40;  
  data[2] = 75;  
  data[3] = 76;  
  data[4] = 90;  
}
```



# Declaring Arrays

```
int[] data = new int[5]; // Declare, create  
void setup() {  
  size(100, 100);  
  data[0] = 19; // Assign  
  data[1] = 40;  
  data[2] = 75;  
  data[3] = 76;  
  data[4] = 90;  
}
```

# Declaring Arrays

```
int[] data = { 19, 40, 75, 76, 90 }; // Declare, create, assign  
void setup() {  
  size(100, 100);  
}
```

# Array Functions :

## Determining Length

```
float[] data = { 19.0, 40.0, 75.0, 76.0, 90.0 };  
void setup() {  
    for (int i = 0; i < data.length; i++) { // For each array element,  
        println(data[i]);  
    }  
}
```

## Code Challenge (20 min)

Prepare an application that prints random words from a selected array of words when mouse is clicked.

i.e.

```
//refer to https://processing.org/reference/text\_.html
```

```
//Let's make bullshit generator
```

```
String[] bullsGen = {"Hello", "you", "here", "there", "door"};
```

```
text(bullsGen[int(random(bullsGen.length))], 20,22);
```



**Let's Have a Break**

## **Code Challenge:**

Let's create a geometric shape with a color set

# Object Oriented Programming:

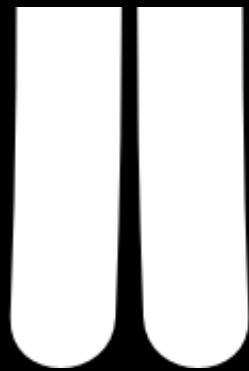
If you've never heard the term before, OOP is just a small conceptual leap in how you think about coding. Rather than giving the machine the task of dealing with a linear progression of instructions, which is what you've done so far, you think in terms of creating objects in memory: self-contained data boxes that you can then use within the familiar linear programming style.

Ref: Generative Art by Matt Pearson





**Let's define a human**



# **Bouncer Example Revisited**

<https://github.com/selcukartut/VA345CreativeCoding>

# Example 2: Code to draw a few circles at the click of the mouse

```
int _num = 10;

void setup() {
    size(500,300);
    background(255);
    smooth();
    strokeWeight(1);
    fill(150, 50);
    drawCircles();
}

void draw() {
}

void mouseReleased() {
    drawCircles();
}
```

```
void drawCircles() {
    for (int i=0; i<_num; i++) {
        float x = random(width);
        float y = random(height);
        float radius = random(100) + 10;
        noStroke();
        ellipse(x, y, radius*2, radius*2);
        stroke(0, 150);
        ellipse(x, y, 10, 10);
    }
}
```

# A Circle class

```
class Circle {  
    float x, y;  
    float radius;  
    color linecol, fillcol;  
    float alph;  
  
    //Constructor  
    Circle () {  
        x = random(width);  
        y = random(height);  
        radius = random(100) + 10;  
        linecol = color(random(255), random(255),  
            random(255));  
        fillcol = color(random(255), random(255),  
            random(255));  
        alph = random(255);  
    }  
}
```

Now, update the **drawCircles** function to call this method after you've created the object

```
void drawCircles() {  
    for (int i=0; i<_num; i++){  
        Circle thisCirc = new Circle();  
        thisCirc.drawMe();  
    }  
}
```