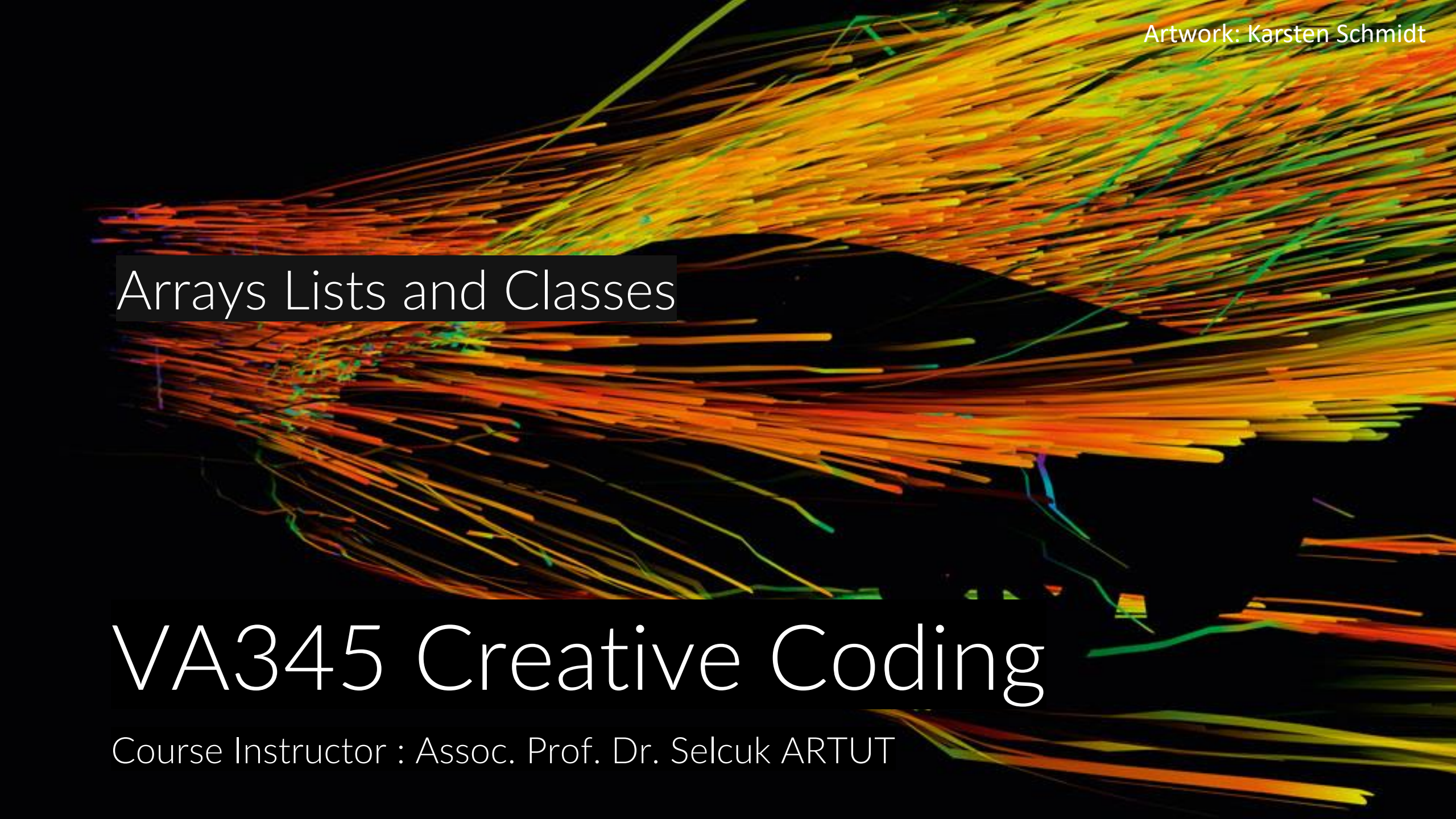


Arrays Lists and Classes

VA345 Creative Coding

Course Instructor : Assoc. 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

Prof. Dr. Miroslaw Majewski

He was worn in Poland, educated in mathematics. M.Sc. and Ph.D. in non-classical geometries at the Nicholas Copernicus University in Poland. Author of about 50 papers and books on computer graphics, applications of computers in education, mathematics, and computer science education, geometry in art and architecture. Some of his recent papers and books are related to the geometry in Islamic art and history of medieval Islamic mathematics. His book 'Islamic Geometric Ornament in Istanbul' shows modern detailed geometric constructions of many geometric ornaments that can be seen in Istanbul. His new series of books 'Sketches on geometry and art' is a systematic overview of various geometric ideas embedded in ancient as well as in modern architecture and architectural decorations. The most recent book in this series (in Polish) contains a detailed review of Central Asian methods in designing geometric patterns used in Islamic art and architecture. Recently he started writing a multipart monograph on decagonal geometric patterns. The first part of this book is completely devoted to Central Asian, XII-XVII century, geometric pattern design methods. The second part of this work is devoted to Seljuk and Ottoman design styles. The third part will cover a structural pattern design methods. Selected chapters of this monograph are printed already by Istanbul Design Publishing, a division of the Istanbul Design Center. More about his works at http://majewski.wordpress.com/.

İstanbul Tasarım Merkezi Geometrik Desenler Okulu Atölyesi

İstanbul'da Geometri ve Desen

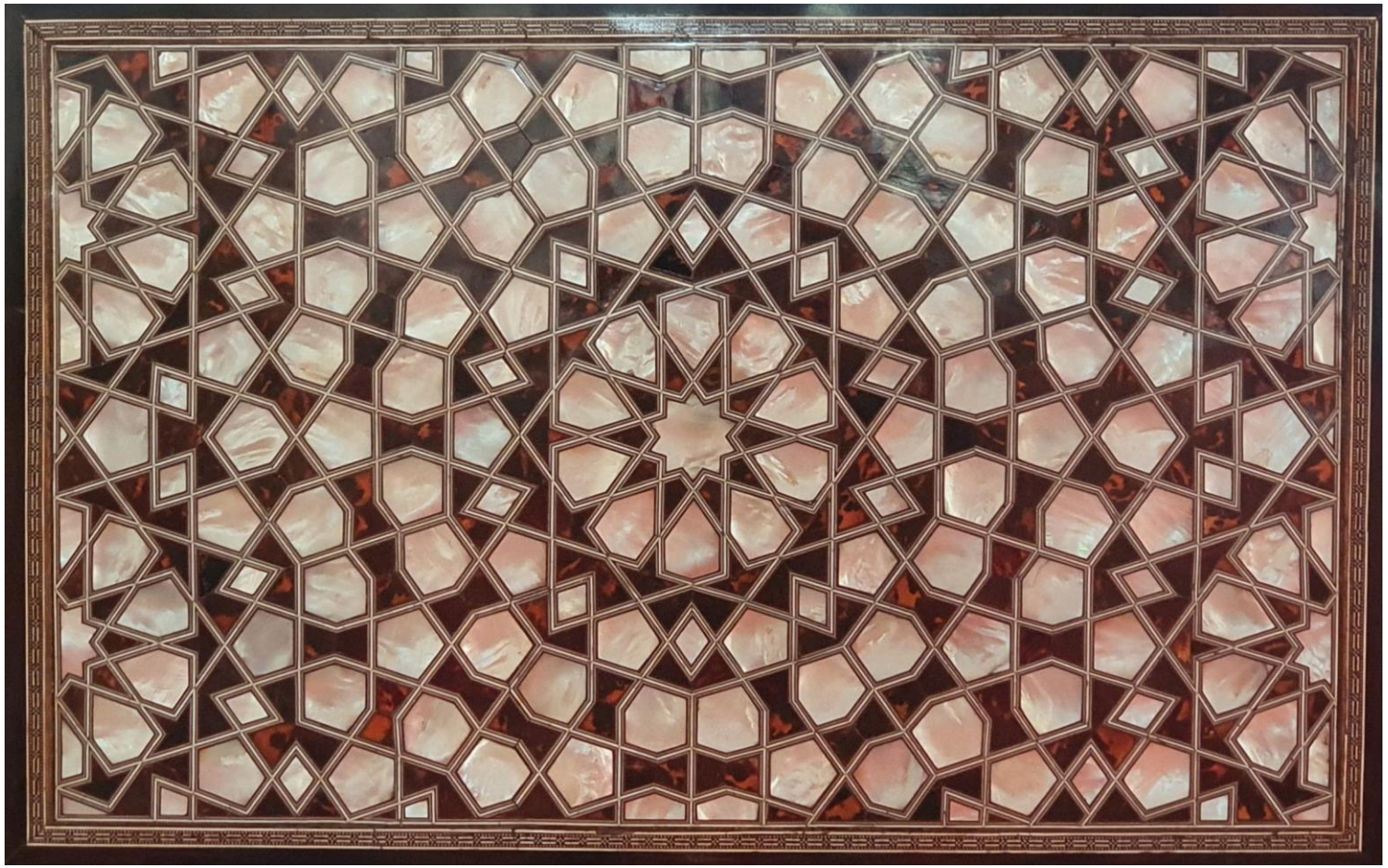
Istanbul Design Center, School of Geometric Patterns Workshop

Geometry and Pattern in Istanbul

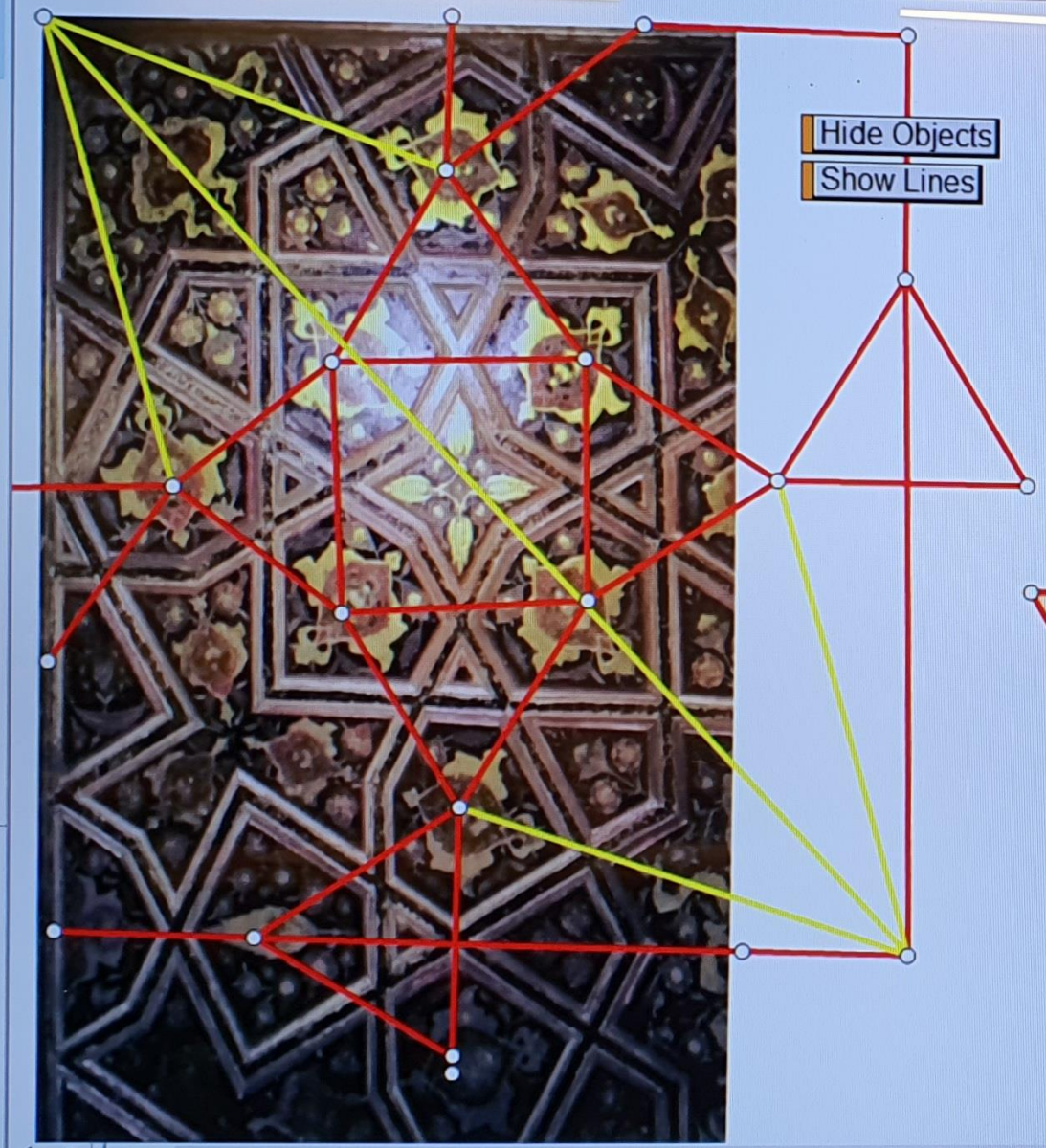
Prof. Dr. Miroslaw L. Majewski

09-13 Kasım / 09-13 November 2021

İSTANBUL TASARIM MERKEZİ | ISTANBUL DESIGN CENTER



File Edit Display Construct Transform Measure Number Graph Window Help



Hide Objects

Show Lines

E

1 2

B I U

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)

https://p5js.org/learn/debugging.html


Let’s Have a Break

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, as 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 demonstation

let xs = [50, 61, 83, 69]

Arrays

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

Array Functions :

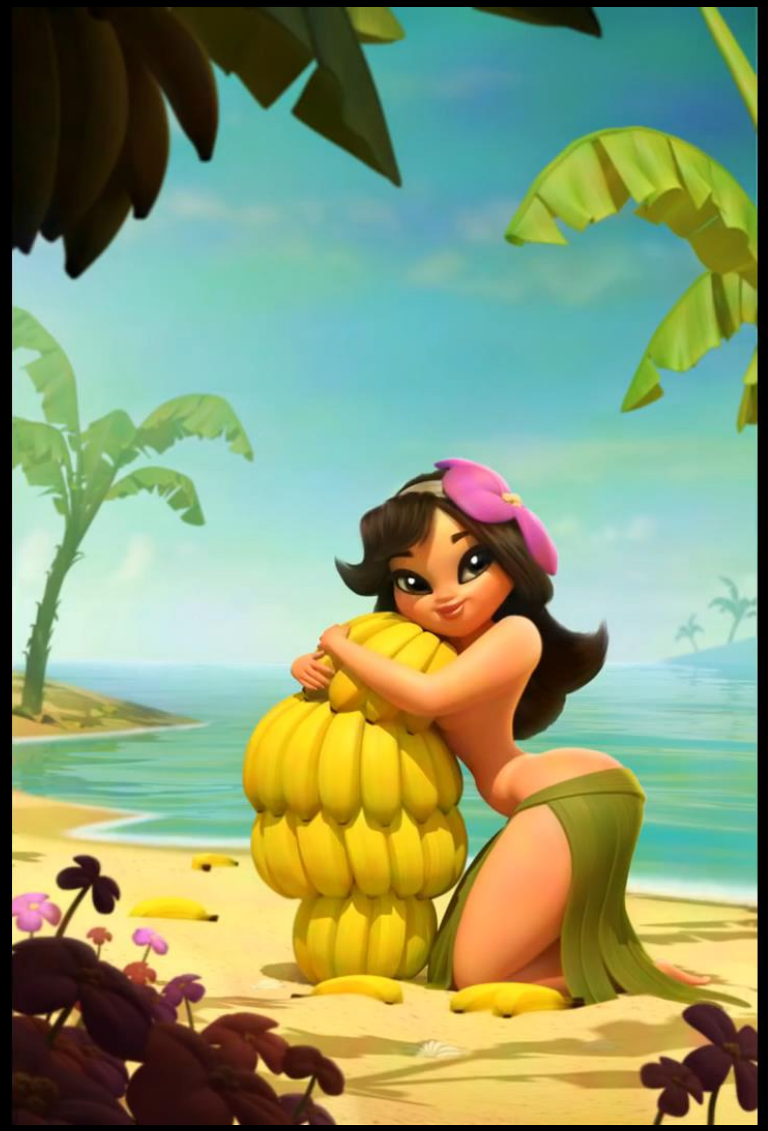
Determining Length

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

Code Challenge (10 min)

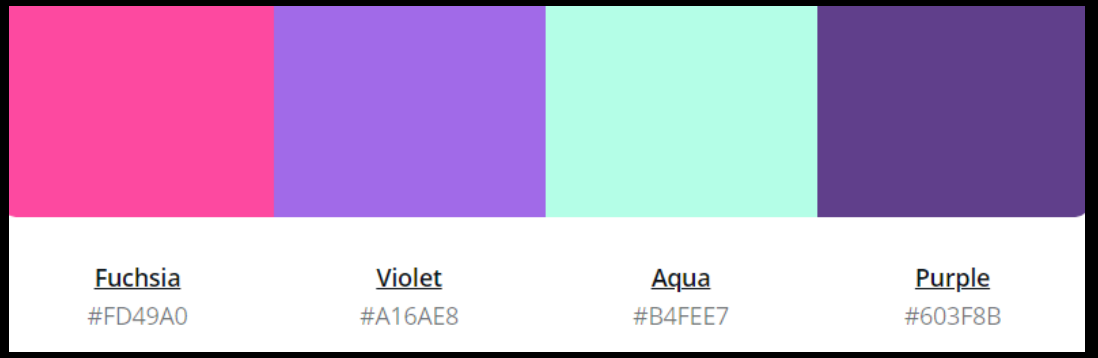
Prepare an application that prints random words from a a selected array of words whenever a mouse is clicked.

Color Theory Recap :



Created by Anders Ehrenborg and Bill Presing
Used with Permission

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



Code Challenge (10 min)

Let’s create an interactive background. When a mouse is clicked, the backround should change to a random color in an array

Refer to : https://www.canva.com/colors/color-palettes/

```
let col = ["#FD49A0", "#A16AE8", "#B4FEE7"];
let bck = col[0];
function setup() {
	createCanvas(400, 400);
}
function draw() {
	background(bck);
}
function mouseClicked(){
	bck = col[int(random(3))];
}
```

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

Code to draw a few circles at the click of the mouse

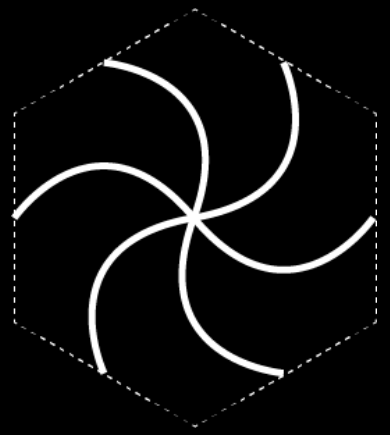
```
let _num = 10;
function setup() {
        createCanvas(500, 300);
}

function draw() {

}

function mousePressed(){
        drawCircles();
}
```

```
function drawCircles() {
    for (let i=0; i<_num; i++) {
        let x = random(width);
        let y = random(height);
        let radius = random(10) + 10;
        noStroke();
        let fillcol = color(random(255), random(255), random(255), random(255));
        fill(fillcol);
        circle(x, y, radius*2);
        }
}
```

A Circle class

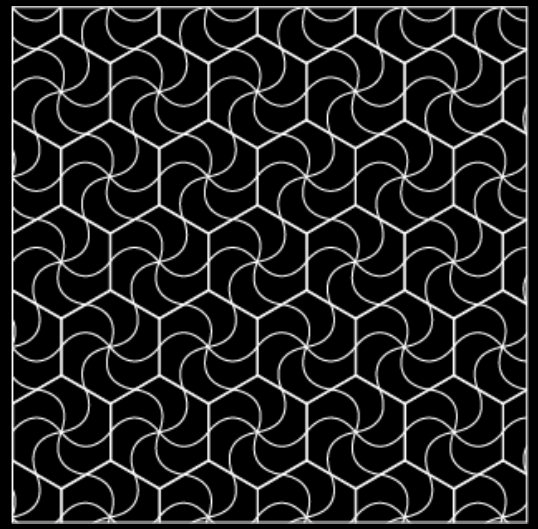
```
let _num = 100;

function setup() {
        createCanvas(500, 300);
}

function draw() {
}


function mousePressed(){
        for (let i = 0; i < _num; i++) {
                let thisCirc = new Circle();
                thisCirc.display();
         }
}
```

```
// Circle class

class Circle {
        constructor() {
                let radius;
                let fillcol;
                this.x = random(width);
                this.y = random(height);
                this.radius = random(10) + 10;
        }

        display() {
                this.fillcol = color(random(255), random(255), random(255), random(255));
                noStroke();
                fill(this.fillcol);
                circle(this.x, this.y, this.radius);
        }
}
```



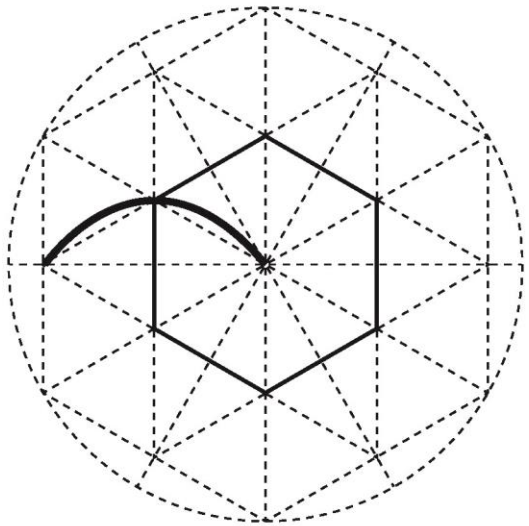
Revisiting

Drawing an Islamic Geometric Pattern / Elhambra, Granda Spain (1302-1391)



Coding an Islamic Geometric Pattern / Elhambra, Granda Spain (1302-1391)

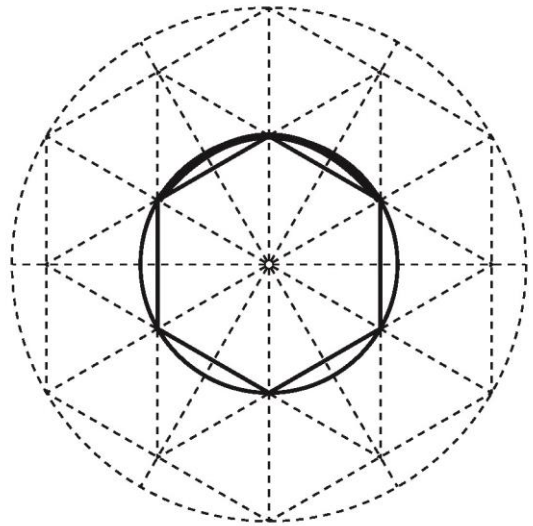
Step 1 : Let's focus on the inner polygon (Refer to Drawing003)



VA345 Creative Coding
Instructor: Selcuk ARTUT

Coding an Islamic Geometric Pattern / Elhambra, Granda Spain (1302-1391)

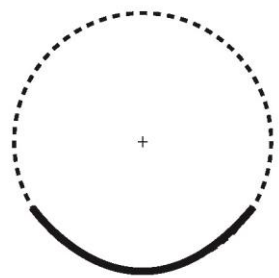
Step 2 : The arc we are looking for is a 1/3 of the circumference circle of the polygon



Coding an Islamic Geometric Pattern / Elhambra, Granda Spain (1302-1391)

Step 3 : Let's use p5.js's arc function
arc(x, y, w, h, start, stop, [mode], [detail])

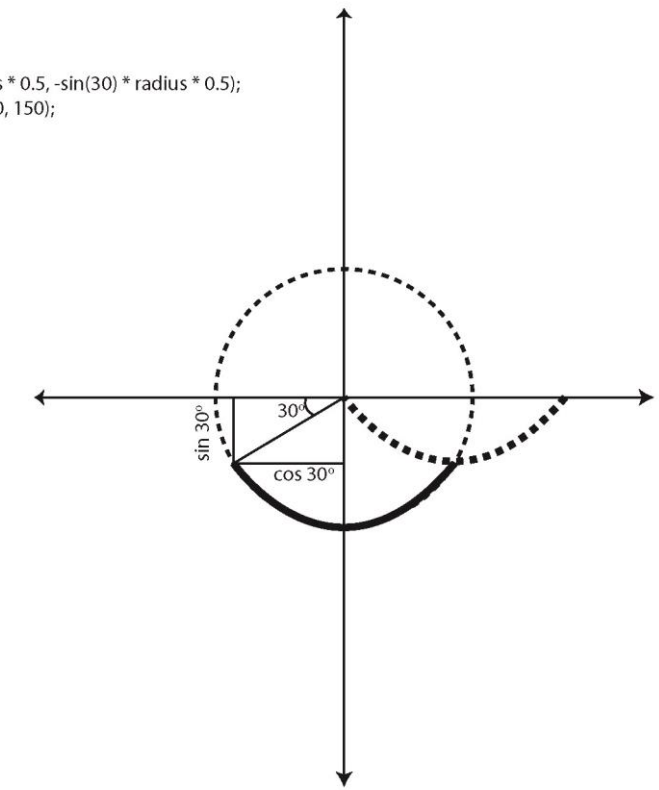
```
let radius = 100;

function setup() {
 createCanvas(400, 400);
 angleMode(DEGREES);
}

function draw() {
 background(0);
 noFill();
 stroke(255);
 arc(0, 0, radius, radius, 30, 150);
}
```



Coding an Islamic Geometric Pattern / Elhambra, Granda Spain (1302-1391)

Step 4 : We need to translate it

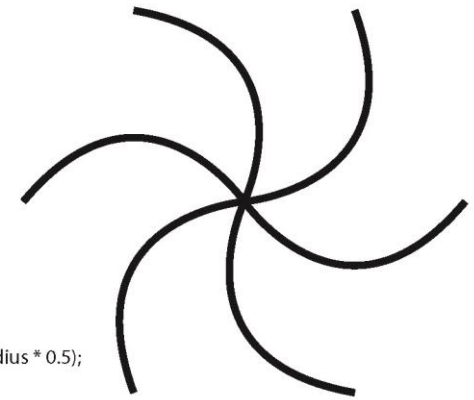
```
let radius = 100;

function setup() {
  createCanvas(400, 400);
  angleMode(DEGREES);
}

function draw() {
  background(0);
  noFill();
  stroke(255);
  push();
  translate(cos(30) * radius * 0.5, -sin(30) * radius * 0.5);
  arc(0, 0, radius, radius, 30, 150);
  pop();
}
```



Coding an Islamic Geometric Pattern / Elhambra, Granda Spain (1302-1391)

Step 5 : Let's copy and rotate six times

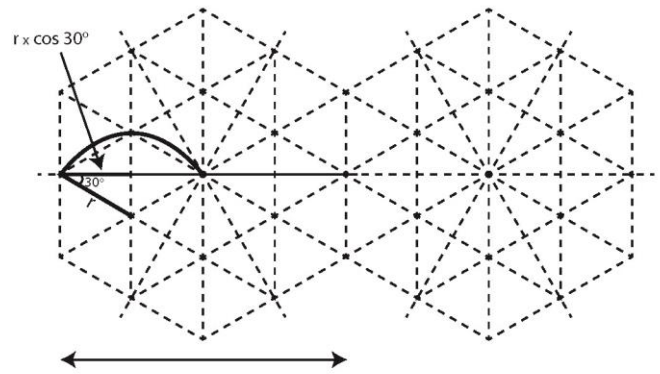
```
let radius = 100;

function setup() {
 createCanvas(400, 400);
 angleMode(DEGREES);
}

function draw() {
 background(0);
 noFill();
 //arc(x, y, w, h, start, stop, [mode], [detail])
 stroke(255);
 for (let i = 0; i < 6; i++) {
  strokeWeight(2);
  push();
  rotate(60*i);
  translate(cos(30) * radius * 0.5, -sin(30) * radius * 0.5);
  arc(0, 0, radius, radius, 30, 150);
  pop();
 }
}
```



Coding an Islamic Geometric Pattern / Elhambra, Granda Spain (1302-1391)

Step 6 : Tesselation

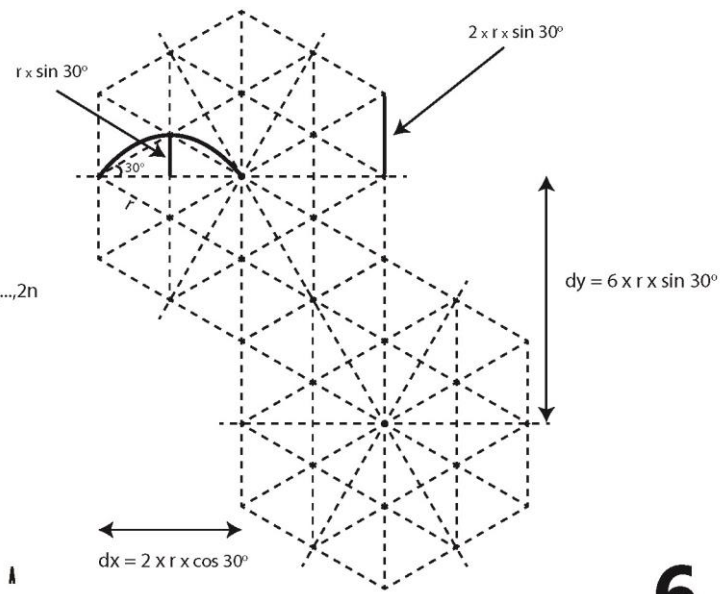
translation in row 1,3,...,2n+1



dx = 4 x r x cos 30°

dy = 0

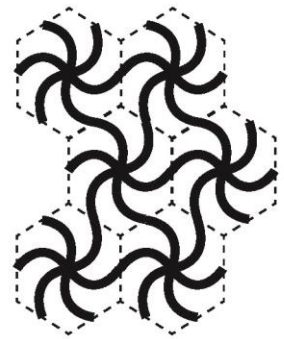
translation in row 2,4,...,2n



Coding an Islamic Geometric Pattern / Elhambra, Granda Spain (1302-1391)

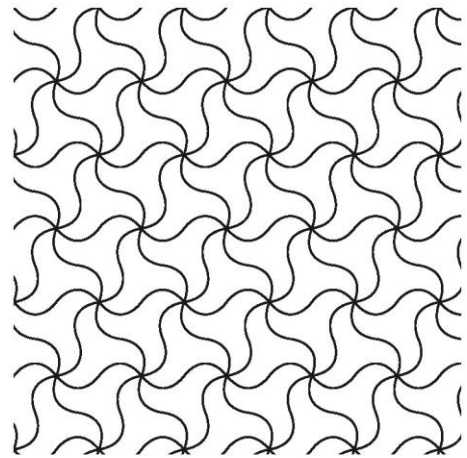
Step 7 : Erase all the construction lines and thicken your pattern's lines

```
// Tile class
class Tile {
  constructor() {
    this.diameter = 50;
    this.cycles = 0;
  }

  display() {

    push();
    let rVal= frameCount % 360;
    if(this.cycles%4) rotate(cos(rVal)*360);
    if(rVal == 0){
      //cycle
      this.cycles ++;
    }


    for (let i = 0; i < 6; i++) {
      push();
      rotate(60 * i);
      translate(cos(30) * this.diameter * 0.5, -sin(30) * this.diameter * 0.5);
      arc(0, 0, this.diameter, this.diameter, 30, 150);
      pop();
    }
    pop();
  }
}
```

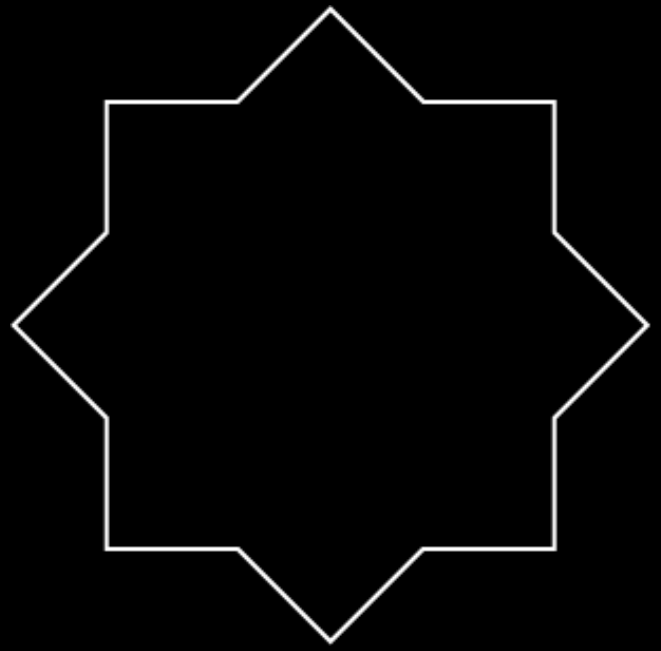


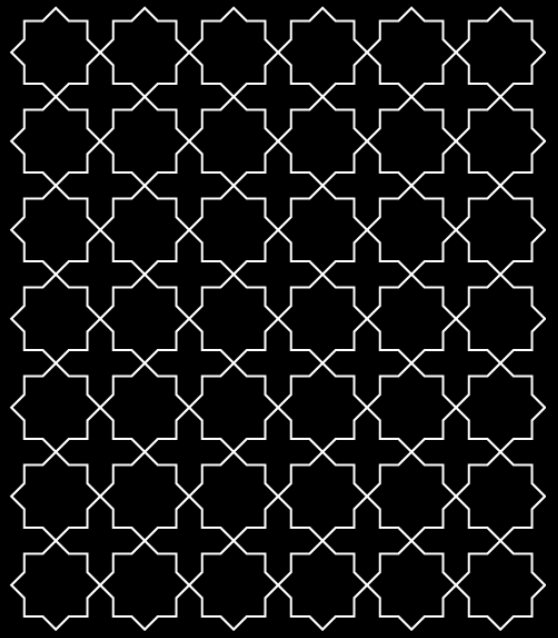
Coding an Islamic Geometric Pattern / Elhambra, Granda Spain (1302-1391)

Step 8 : Tesselate



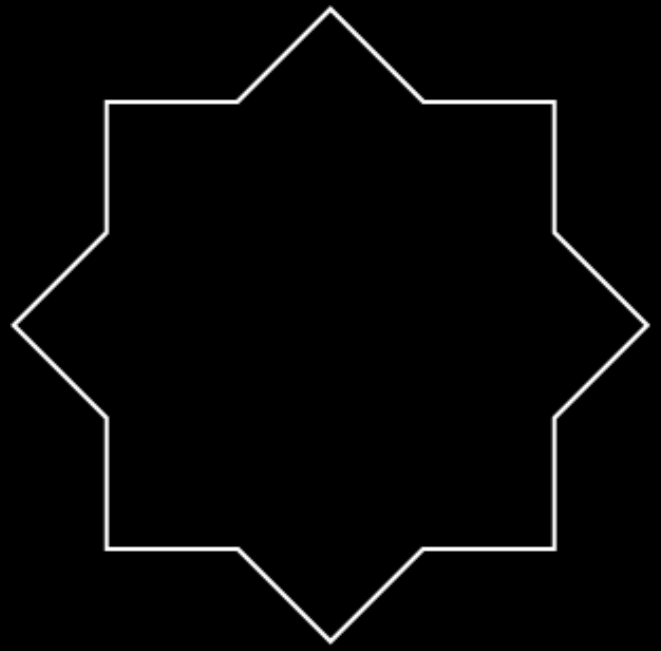
```
let radius = 25;
let tiles = []; // Declare array
let nRow = 9;
let nCol = 9;

function setup() {
  createCanvas(693, 600);
  angleMode(DEGREES);
  for (let i = 0; i < nRow * nCol; i++) {
    tiles.push(new Tile());
  }
}

function draw() {
  background(0);
  noFill();
  stroke(0);
  for (let r = 0; r < nRow; r++) {
    for (let c = 0; c < nCol; c++) {
      push();
      if(r%2){
        //rows 0,2,4,6
        translate(c * cos(30) * radius * 4,  radius * r * 6 * sin(30));
      }else{
        //rows 1,3,5,7
        translate(cos(30) * radius * 2 + c * cos(30) * radius * 4, radius * r * 6 * sin(30));
      }
      tiles[r+c*nRow].display();
      pop();
    }
  }
}
```

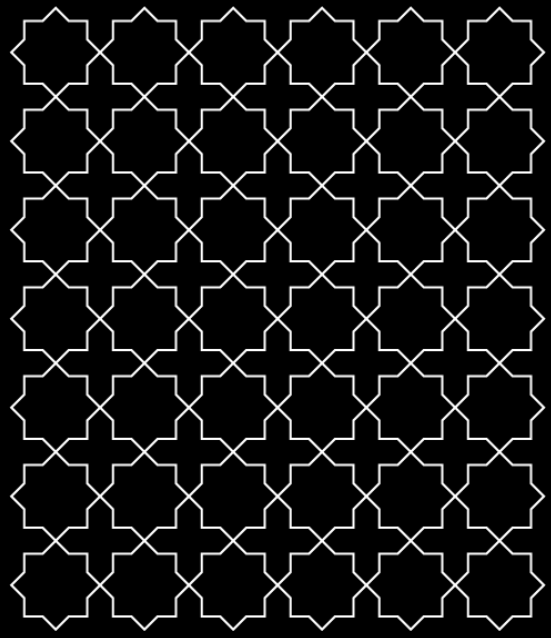




Drawing in Class

Drawing an Islamic Geometric Pattern / Cappella Palatina





Assignment 005

Code an Islamic Geometric Pattern / Cappella Palatina

Use the construction motif as a class member to tesselate the pattern