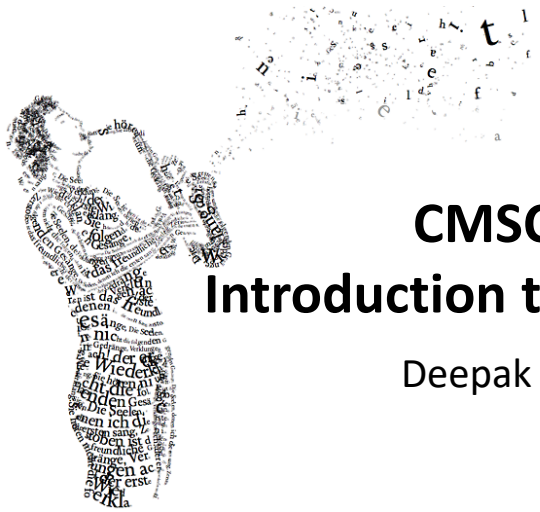
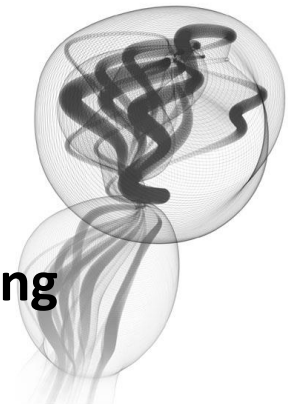


CMSC109
Introduction to Computing

Deepak Kumar



1

Administrivia

CMSC109: Introduction to Computing

Fall 2022

Course Website: https://cs.brynmawr.edu/Courses/cs109/fall2022/
Instructor:

Deepak Kumar, (dkumar@brynmawr.edu)

Lectures

MoWe 1:10p to 2:30p in Park 245

TA-Support

>20 hrs/week in Park 231

Labs – Attendance is required

- Mondays 2:40p to 4:00p in Park 231

Office Hours

TBA

Grading

• Assignments	25%
• Exam 1	20%
• Exam 2	20%
• Exam 3	25%
• Lab Attendance	10%
Total	100%

2

2

Administrivia

Software

Processing 4.X

- Already installed in the CS Lab
- Also available for your own computer @ www.processing.org
- Processing == Java

Required

No text is required. We will provide online materials and handouts.

Dropbox Account: Please go to dropbox.com and register. You will be using dropbox to submit many of your assignments. You will need to have this set up by the end of Week#1.

3

3

Class Lottery

- Make sure to sign-in your name.
- If you are not on the class list, sign on the attached sheet. We will contact you by e-mail as soon as we have confirmation from other students.

4

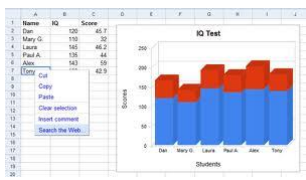
4

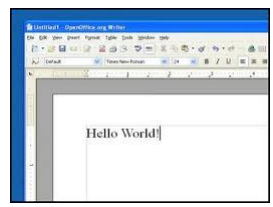
What is Computing?

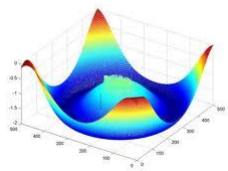
5

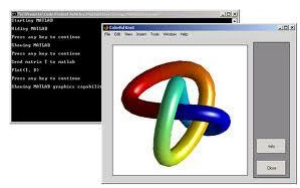
5

Computing: Your Parent's View









6

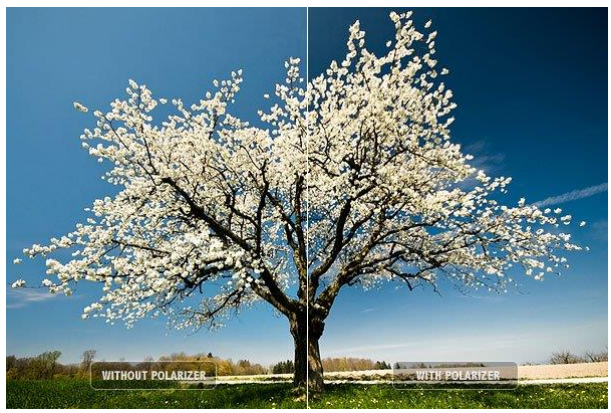
6

Computing: internet, e-mail, network…

7

7

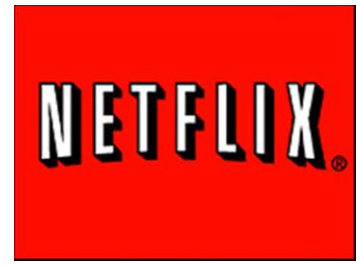
Computing: Digital Photography



8

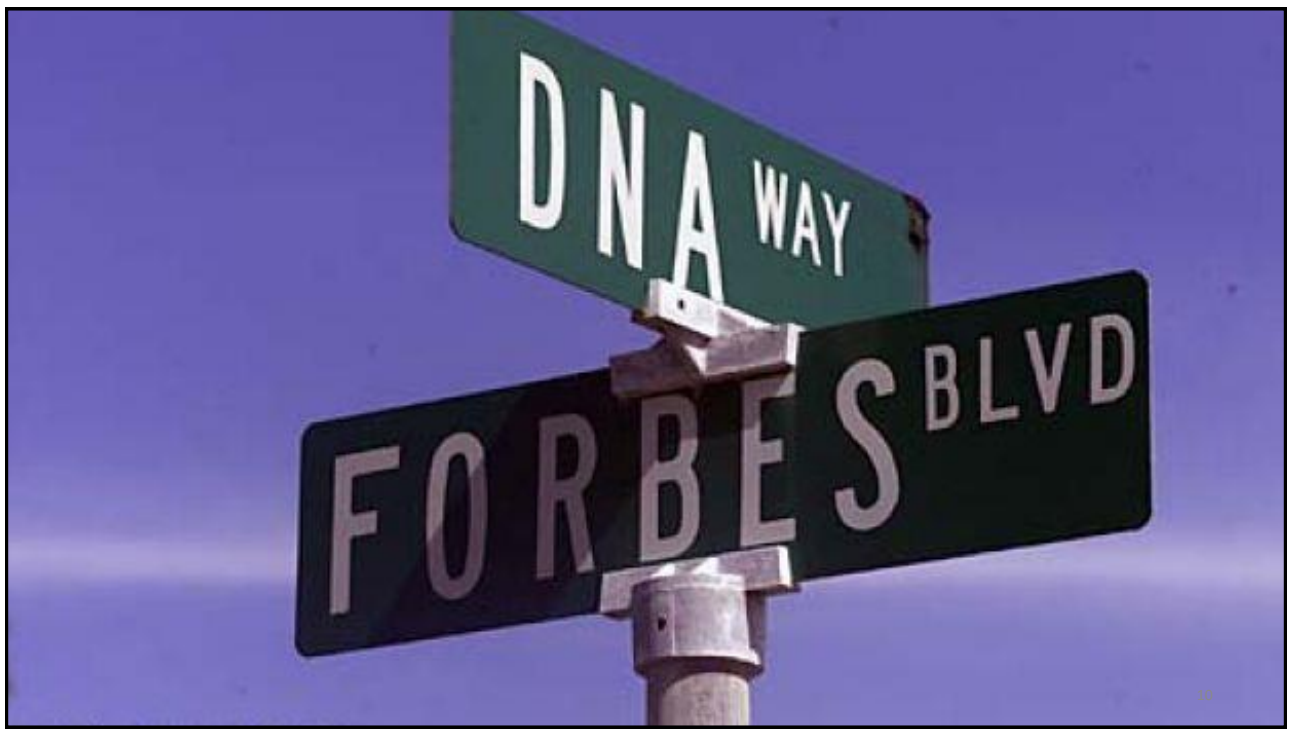
8

Computing: Entertainment…



9

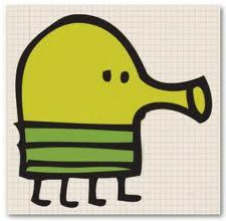
9

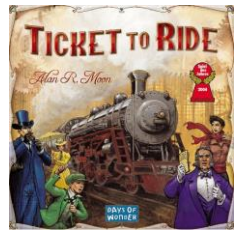


10

10

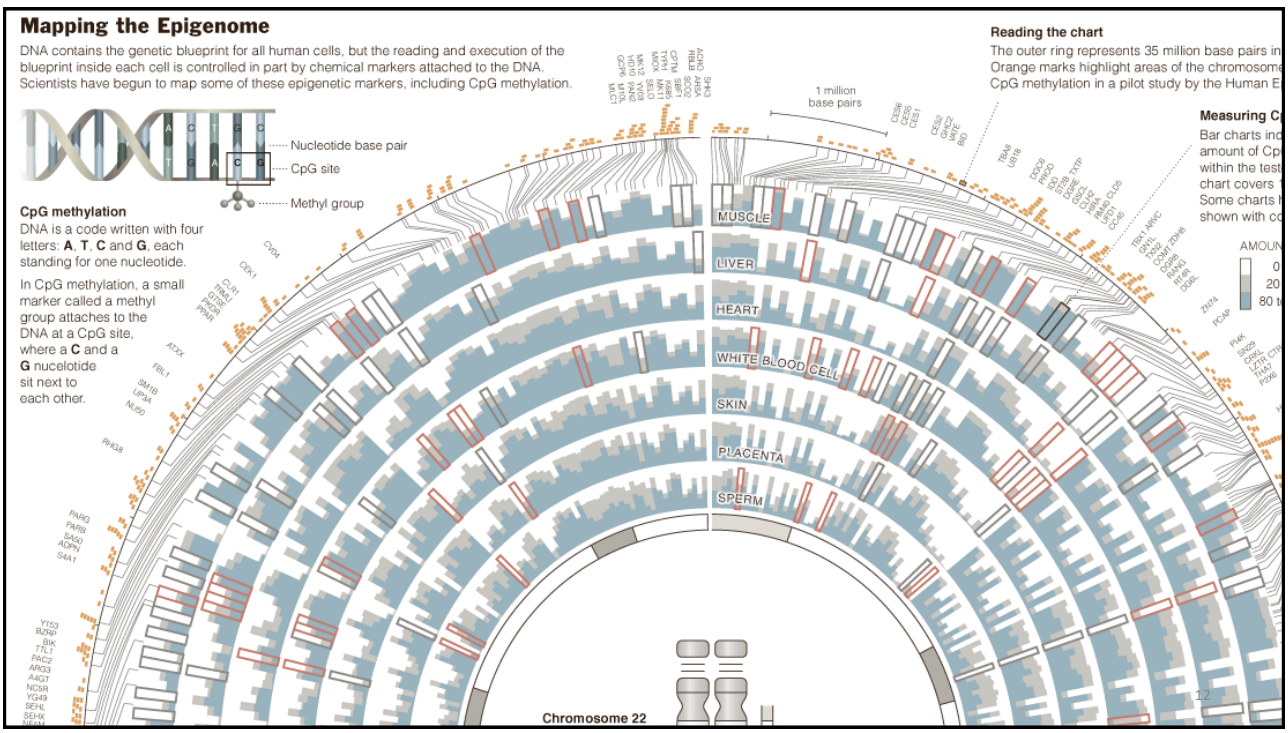
Computing: Entertainment…





11

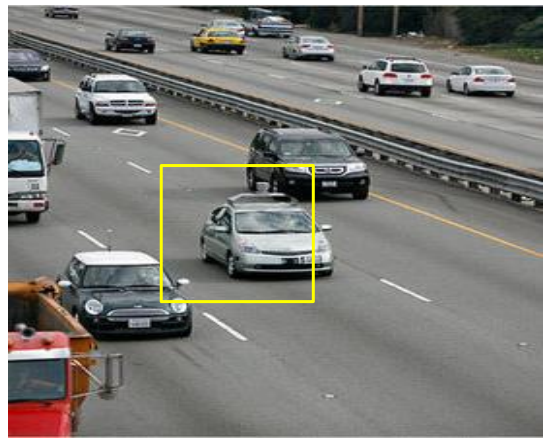
11



12

12

Self-driving (Autonomous) Cars



13

13

Some Areas in Computer Science

Artificial Intelligence

Robotics

Human-Computer Interaction

Computer Graphics

Computer Vision

Operating Systems

Computer Networking

Databases

Computer Security

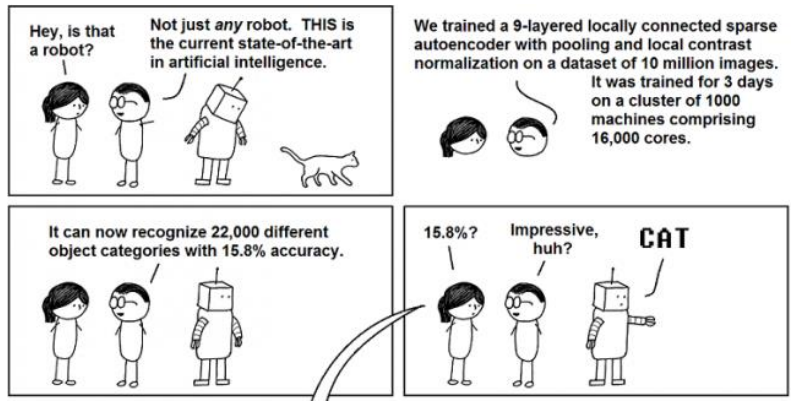
Ubiquitous Computing

14

14

More trendy…

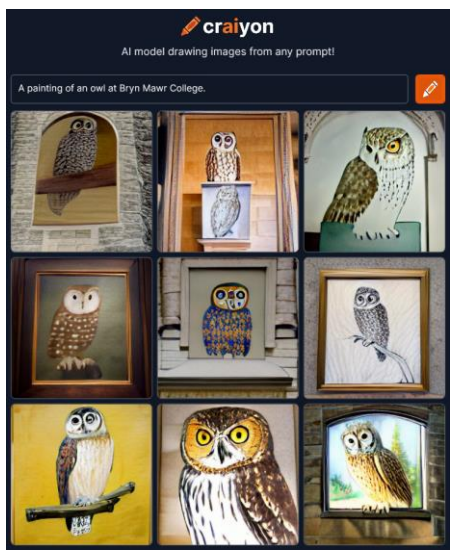
- Machine Learning (Deep Learning)
- Data Science (Big Data)
- Cybersecurity



15

15

Dall-e: Hot off the presses…

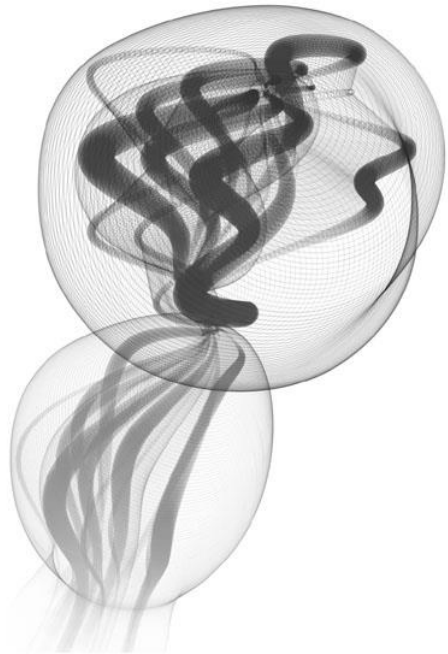


Play here:
https://www.craiyon.com/

16

16

ART



Protobytes
By Ira Greenberg

17

17

What is Computer Science?

Computer science is the study of solving problems using computation

- Computers are part of it, but the emphasis is on the problem solving aspect

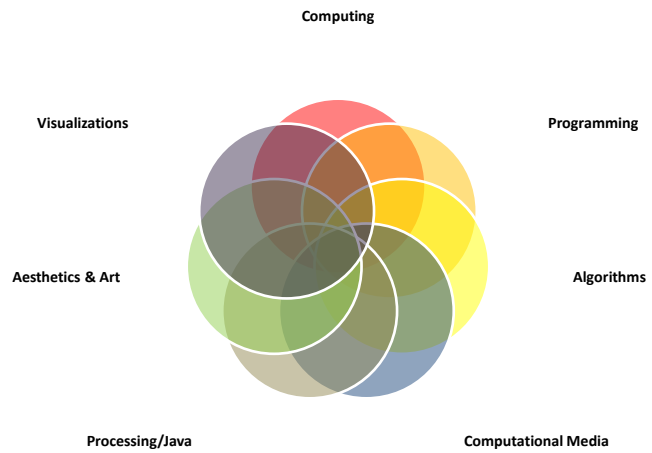
Computer scientists work across disciplines:

Mathematics	Geoscience	Medicine/Surgery
Biology (bioinformatics)	Archaeology	Engineering
Chemistry	Psychology	Linguistics
Physics	Sociology	Art
Geology	Cognitive Science	…

18

18

Creative
Introduction to ^ Computing



19

19

Algorithms

An **algorithm** is an effective method for solving a problem expressed as a finite sequence of instructions. For example,

Put on shoes
- left sock
- right sock
- left shoe
- right shoe

20

20

Exercise: How to draw an owl???

Exercise: How to draw an owl???

- What did we need?
- ***Primitive Shapes***: lines, circles, ovals, rectangles, squares, curves, etc. (also colors)
- Step-by-step Instructions (i.e. ***an algorithm***)
  do this
  then do this
  then this
  etc.

Programming = Writing Apps

Programming is the process of designing, writing, testing, debugging / troubleshooting, and maintaining the source code of computer programs.

This source code is written in a **programming language**.

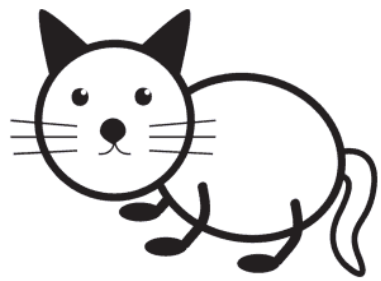
A program codes the steps of an **algorithm** using primitives.

23

23

Algorithm to draw a cat?

Draw a face
Draw ears
Draw eyes
Draw nose and mouth
Draw whiskers
Draw body
Draw legs and feet
Draw tail

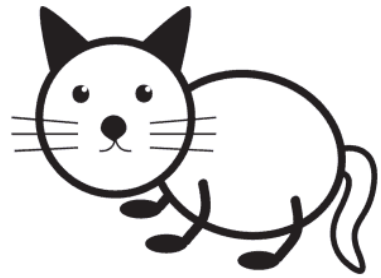


24

24

Algorithm to draw a cat? – Identify Primitives

Draw a face [circle]
Draw ears [triangle]
Draw eyes [circle]
Draw nose and mouth [circle, arc]
Draw whiskers [line]
Draw body [oval]
Draw legs and feet [line, oval]
Draw tail [curve??]



25

25

From Algorithm to Program

- ***Programming Language***
  provides a formal language to write steps
  provides basic primitive operations
  Algorithm is coded into a ***program***

- There are over 3000 programming languages
  Python, C, Java, C++, C#, Visual Basic, JavaScript, PHP, Swift, Go, etc.

- In this class we will learn ***Processing*** (Java)

26

26

A program

```
int areaOfCircle(int radius){
  return PI*radius*radius;
}

r = 10;
area = areaOfCircle(r);
```

27

Programming Languages

Processing/Java/C/C++	Python	Lisp

`int areaOfCircle(int radius){`
`  return PI*radius*radius;`
`}`
 	
`r = 10;`
`area = areaOfCircle(r);`
	`def areaOfCircle(radius):`
	`  return PI*radius*radius;`
	
	`r = 10`
	`area = areaOfCircle(r)`
		`(defun areaOfCircle (radius)`
		`  (return (* PI radius radius)))`
		
		`(setq r 10)`
		`(setq area (areaOfCircle r))`

28

Programming Languages

Processing	Python	Lisp

`int areaOfCircle(int radius){`
`  return PI*radius*radius;`
`}`
 	
`r = 10;`
`area = areaOfCircle(r);`
	`def areaOfCircle(radius):`
	`  return PI*radius*radius;`
	
	`r = 10`
	`area = areaOfCircle(r)`
		`(defun areaOfCircle (radius)`
		`  (return (* PI radius radius)))`
		
		`(setq r 10)`
		`(setq area (areaOfCircle r))`

Python, C, Java, C++, C#, Visual Basic, JavaScript, PHP, Swift, Go, etc.

There are over 3000 of them!

29

29

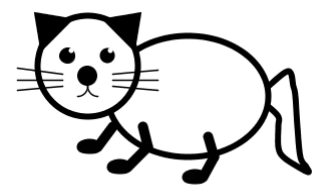
A Processing Program to Draw a Cat

```
void draw() {

  // Body
  strokeWeight(6);
  ellipse(200, 130, 160, 120);

  // Face
  strokeWeight(6);
  ellipse(100, 100, 100, 100);

  // Ears
  // left ear
  strokeWeight(6);
  fill(0);
  beginShape();
  vertex(55, 80);
  vertex(50, 50);
  vertex(80, 55);
  endShape(CLOSE);

  // right ear
  strokeWeight(6);
  fill(0);
  beginShape();
  vertex(120, 55);
  vertex(150, 50);
  vertex(145, 80);
  endShape(CLOSE);
```

```
  // Eyes
  strokeWeight(1);
  fill(0);
  // left eye
  ellipse(80, 90, 15, 15);
  fill(255);
  noStroke();
  ellipse(84, 85, 5, 5);

  // right eye
  strokeWeight(1);
  fill(0);
  ellipse(120, 90, 15, 15);
  fill(255);
  noStroke();
  ellipse(123, 85, 5, 5);

  // Nose and Mouth
  // nose
  fill(0);
  ellipse(100, 110, 20, 20);

  // Whiskers
  stroke(0);
  strokeWeight(1.5);

  //right
  line(125, 108, 170, 104);
  line(125, 114, 170, 114);
  line(125, 120, 170, 124);

  // left
  line(75, 108, 30, 102);
  line(75, 114, 30, 114);
  line(75, 120, 30, 124);
```

```
  // mouth
  stroke(1);
  noFill();
  //ellipse(90, 120, 20, 20);
  arc(90, 120, 20, 20, 0, PI/2);
  //ellipse(110, 120, 20, 20);
  arc(110, 120, 20, 20, PI/2, PI);


  // Legs and feet
  strokeWeight(10);
  line(125, 160, 110, 180);

  line(155, 160, 160, 180);
  line(160, 180, 140, 200);

  line(220, 170, 226, 190);
  line(226, 190, 215, 210);

  fill(0);
  stroke(1);
  ellipse(102, 180, 15, 5);
  ellipse(132, 200, 15, 5);
  ellipse(207, 210, 15, 5);
```

```
  // Legs and feet
  strokeWeight(10);
  line(125, 160, 110, 180);
  line(155, 160, 160, 180);
  line(160, 180, 140, 200);
  line(220, 170, 226, 190);
  line(226, 190, 215, 210);
  fill(0);
  stroke(1);
  ellipse(102, 180, 15, 5);
  ellipse(132, 200, 15, 5);
  ellipse(207, 210, 15, 5);

  // Tail
  noFill();
  strokeWeight(6);
  beginShape();
  curveVertex(280, 120);
  curveVertex(280, 120);
  curveVertex(300, 105);
  curveVertex(305, 120);
  curveVertex(307, 130);
  curveVertex(309, 160);
  curveVertex(312, 190);
  curveVertex(318, 200);
  curveVertex(320, 205);
  curveVertex(310, 200);
  curveVertex(295, 190);
  curveVertex(290, 180);
  curveVertex(290, 160);
  curveVertex(290, 150);
  curveVertex(280, 140);
  curveVertex(280, 140);
  endShape();
} // draw()
```



30

30

How to draw an owl…

```
void draw() {
  // Body
  fill(0);
  textSize(200);
  text("(    )", 100, 300);

  // Eyes
  ellipse(200, 200, 20, 20);
  ellipse(280, 200, 20, 20);

  // Nose/Beek
  strokeWeight(5);
  line(240, 220, 225, 235);
  line(240, 220, 255, 235);

  // Head/Hair
  pushMatrix();
  translate(300, 170);
  rotate(3*PI/2);
  text("{", 0, 0);
  popMatrix();

  // Bryn Mawr
  textSize(24);
  text("BRYN MAWR", 180, 340);
} // draw()
```

31

31

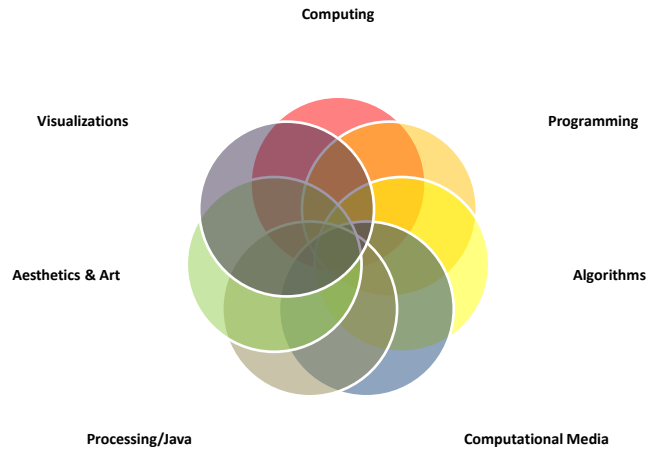
Our Goal

- Use computing to realize works of “art”
- Explore new metaphors from computing: images, animation, interactivity, visualizations
- Learn the basics of computing
- Have fun doing all of the above!

32

32

Creative
Introduction to ^ Computing



33

33

Let's get started…

34

34

Administrivia

Software

Processing 4.x

- Already installed in the CS Lab
- Also available for your own computer @ www.processing.org
- Processing == Java

Required

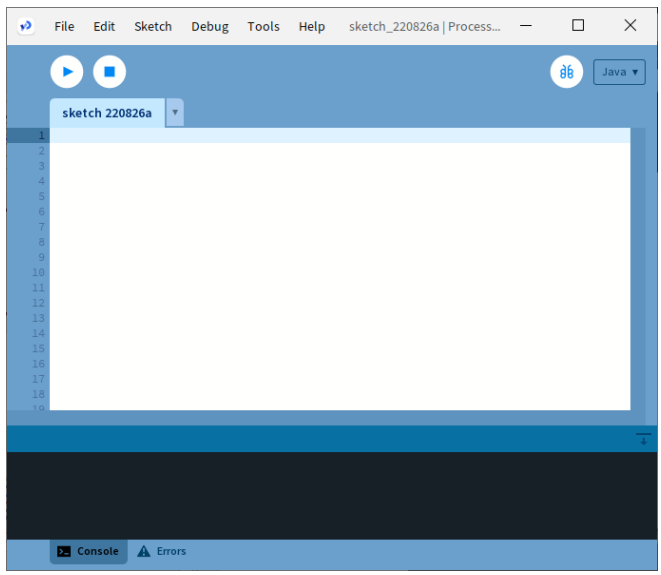
Text: No need to purchase a text.

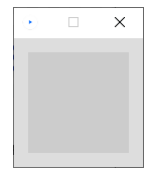
Dropbox Account: Please go to dropbox.com and register. You will be using dropbox to submit many of your assignments.

35

35

Processing 4 IDE

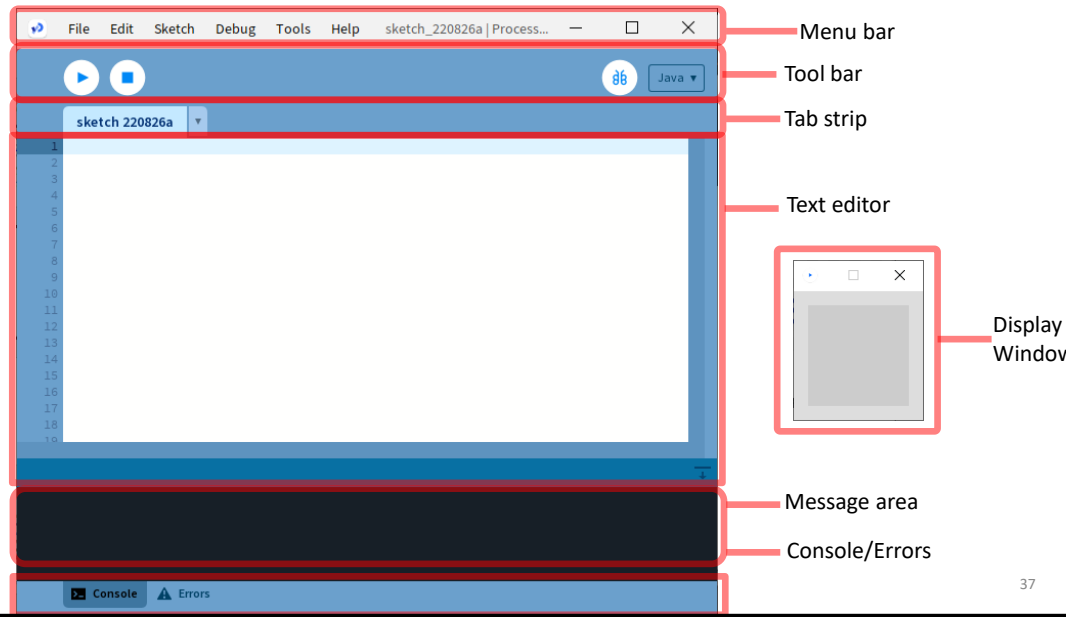




36

36

Processing 4 IDE



37

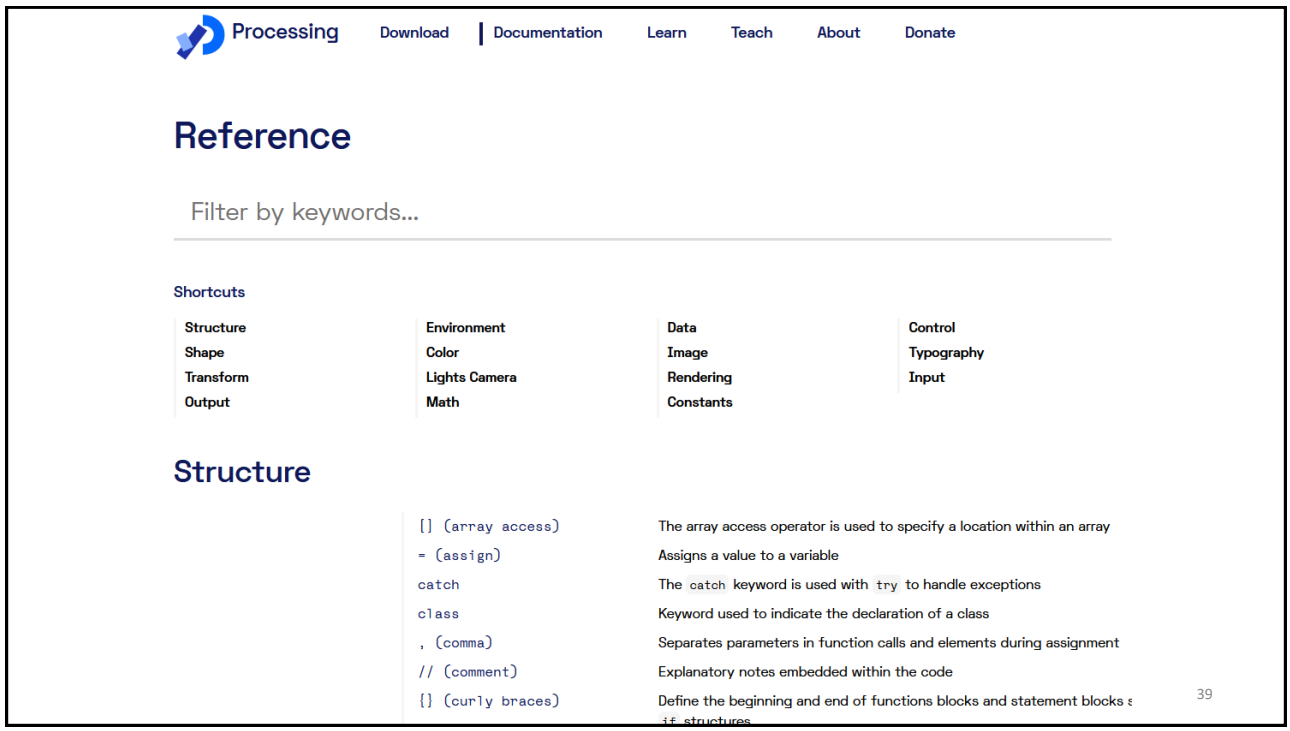
37

Primitive 2D Shapes

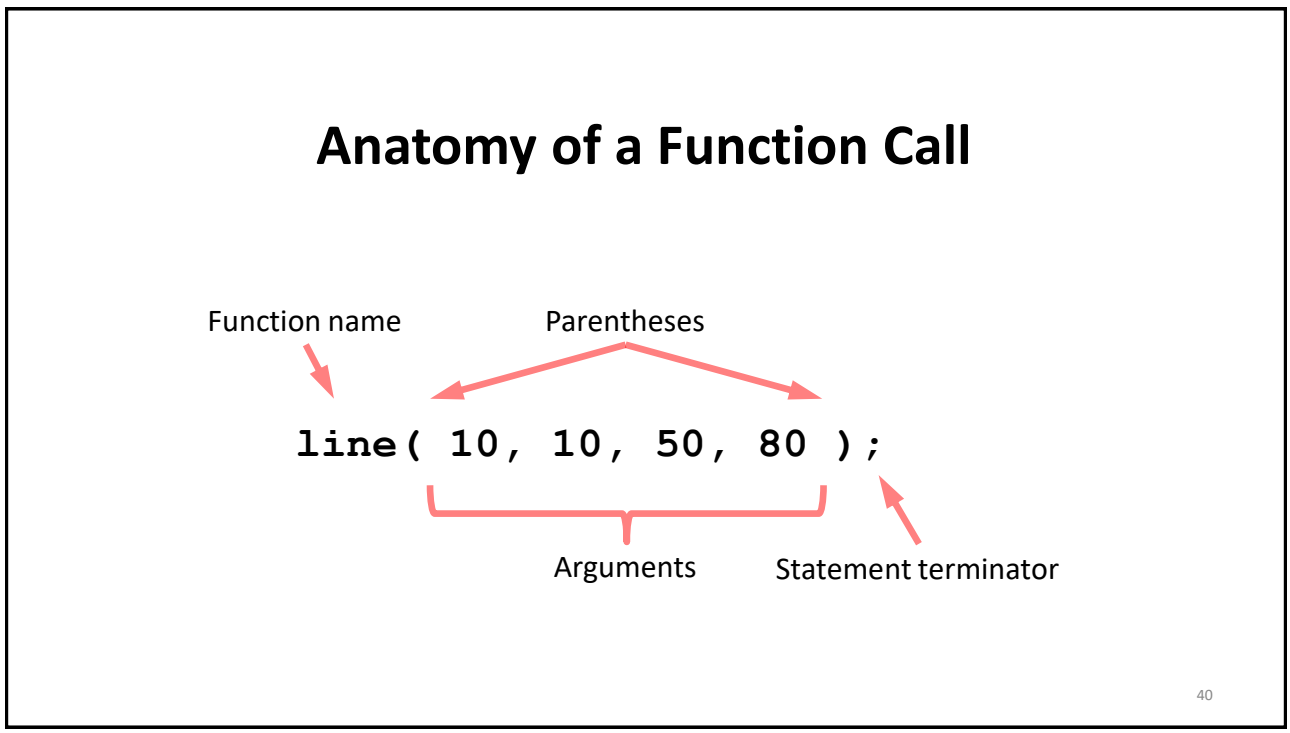
- point
- line
- triangle
- rect (rectangle)
- quad (quadrilateral, four-sided polygon)
- ellipse
- arc (section of an ellipse)
- curve (Catmull-Rom spline)
- bezier (Bezier curve)

38

38

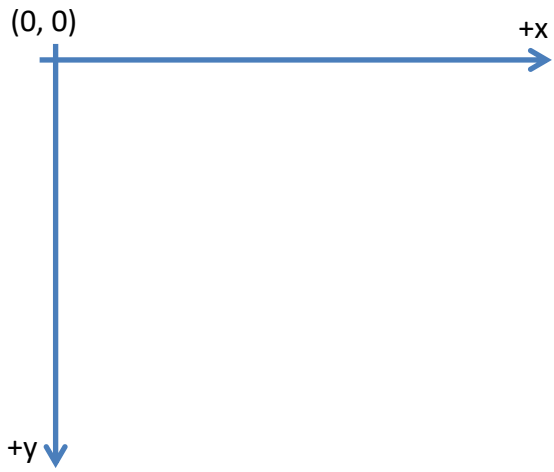

Processing
Download
Documentation
Learn
Teach
About
Donate
Reference
Filter by keywords...
Shortcuts
Structure
Shape
Transform
Output
Environment
Color
Lights Camera
Math
Data
Image
Rendering
Constants
Control
Typography
Input
Structure
[] (array access)
The array access operator is used to specify a location within an array
= (assign)
Assigns a value to a variable
catch
The catch keyword is used with try to handle exceptions
class
Keyword used to indicate the declaration of a class
, (comma)
Separates parameters in function calls and elements during assignment
// (comment)
Explanatory notes embedded within the code
{} (curly braces)
Define the beginning and end of functions blocks and statement blocks
39

39


Anatomy of a Function Call
Function name
Parentheses
line(10, 10, 50, 80);
Arguments
Statement terminator
40

40

Coordinate System



41

41

Pixels

42

42

Processing Canvas

size(*width*, *height* **);**

Set the size of the canvas.

background(*[0..255]* **);**

Set the background grayscale color.

43

Drawing Primitives

```
point( x, y );

line( x1, y1, x2, y2 );

triangle( x1, y1, x2, y2, x3, y3 );

quad( x1, y1, x2, y2, x3, y3, x4, y4 );

rect( x, y width, height );

ellipse( x, y, width, height );
```

44

Colors

Composed of four elements:

1. Red
2. Green
3. Blue
4. Alpha (Transparency)

45

45

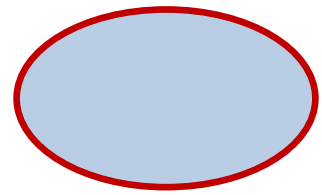
Why 0 .. 255?

46

46

Shape Formatting

1. Fill color
2. Line thickness
3. Line color



These are properties of your paintbrush, not of the object you are painting.

47

47

Fill Color

```
fill(gray);
fill(gray, alpha);
fill(red, green, blue);
fill(red, green, blue, alpha);

noFill();
```

48

48

Stroke (Line) Color

```
stroke(gray);
stroke(gray, alpha);
stroke(red, green, blue);
stroke(red, green, blue, alpha);

noStroke();
```

49

strokeCap()

```
smooth();
strokeWeight(12.0);
strokeCap(ROUND);
line(20, 30, 80, 30);
strokeCap(SQUARE);
line(20, 50, 80, 50);
strokeCap(PROJECT);
line(20, 70, 80, 70);
```

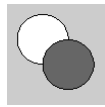
strokeWeight()

```
smooth();
strokeWeight(1);   // Default
line(20, 20, 80, 20);
strokeWeight(4);   // Thicker
line(20, 40, 80, 40);
strokeWeight(10);  // Beastly
line(20, 70, 80, 70);
```

http://processing.org/reference/strokeCap_.html
http://processing.org/reference/strokeWeight_.html

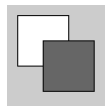
50

ellipseMode



```
ellipseMode(CENTER);
ellipse(35, 35, 50, 50);
ellipseMode(CORNER);
fill(102);
ellipse(35, 35, 50, 50);
```

rectMode

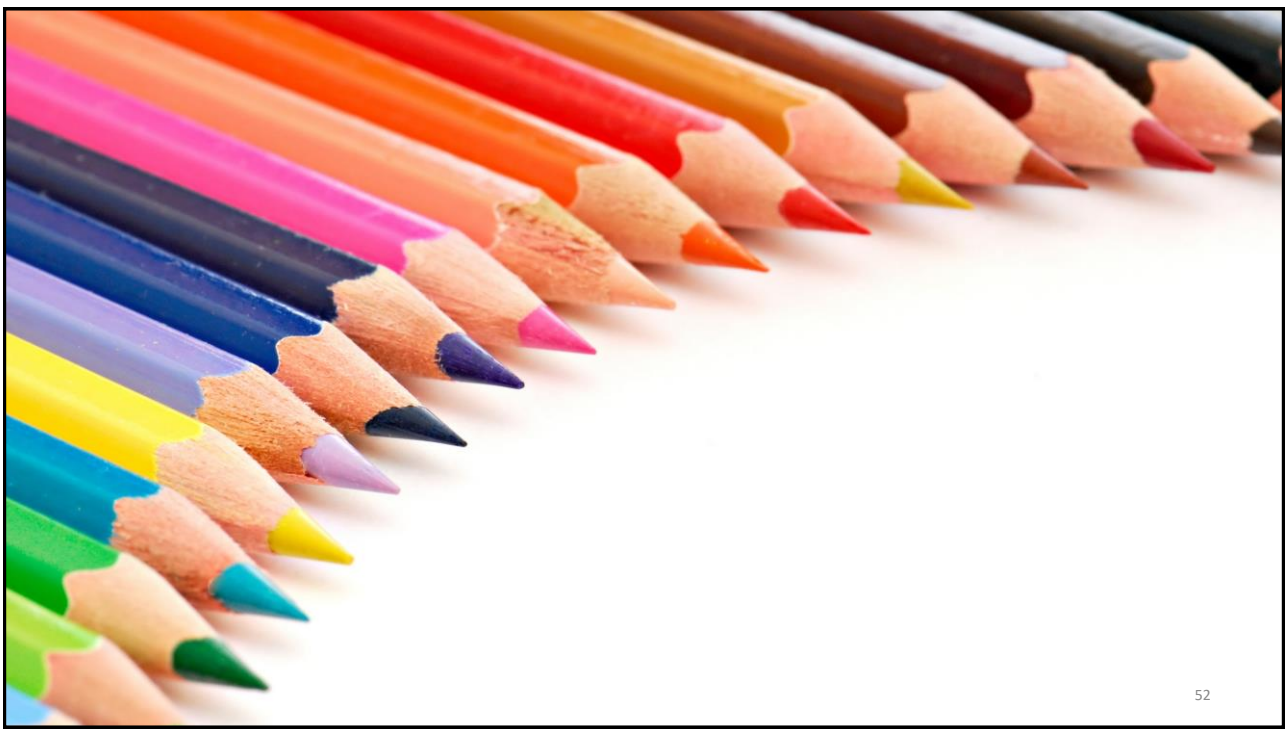


```
rectMode(CENTER);
rect(35, 35, 50, 50);
rectMode(CORNER);
fill(102);
rect(35, 35, 50, 50);
```

http://processing.org/reference/ellipseMode_.html
http://processing.org/reference/rectMode_.html

51

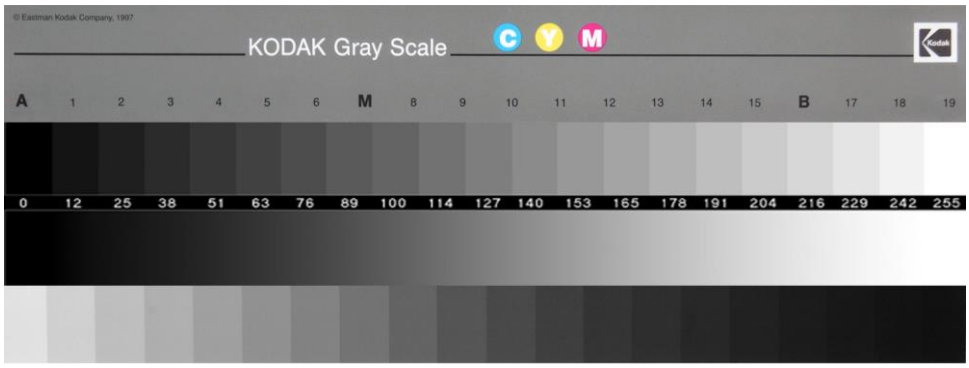
51



52

52

256 Shades of Gray



53