CMSC109
Introduction to Computing
Deepak Kumar

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%
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.

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.
What is Computing?

Computing: Your Parent’s View

![Image of Excel chart]

![Image of programming output]

![Image of 3D graph]

![Image of software interface]
Computing: internet, e-mail, network...

Computing: Digital Photography
Computing: Entertainment...

Mapping the Epigenome

DNA contains the genetic blueprint for all human cells, but the reading and execution of the blueprint inside each cell is controlled in part by chemical markers attached to the DNA. Scientists have begun to map some of those epigenetic markers, including CpG methylation.

CpG methylation

DNA is a code written with four letters: A, T, C, and G, each standing for one nucleotide. In CpG methylation, a small marker called a methyl group attaches to the DNA at a CpG site, where a C and a G nucleotide sit next to each other.

Reading the chart

The outer ring represents 35 million base pairs. Orange marks highlight areas of the chromosome. CpG methylation in a pilot study by the Human E
Self-driving (Autonomous) Cars

Some Areas in Computer Science

- Artificial Intelligence
- Robotics
- Human-Computer Interaction
- Computer Graphics
- Computer Vision
- Operating Systems
- Computer Networking
- Databases
- Computer Security
- Ubiquitous Computing
More trendy...

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

Dall-e: Hot off the presses...

Play here: https://www.craiyon.com/
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
- Biology (bioinformatics)
- Chemistry
- Physics
- Geology
- Geoscience
- Archaeology
- Psychology
- Sociology
- Cognitive Science
- Medicine/Surgery
- Engineering
- Linguistics
- Art
- ...
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
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.

---

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
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??]

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)
A program

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

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

Programming Languages

<table>
<thead>
<tr>
<th>Processing/Java/C/C++</th>
<th>Python</th>
<th>Lisp</th>
</tr>
</thead>
</table>
| int areaOfCircle(int radius){
    return PI*radius*radius;
}                               | def areaOfCircle(radius):
    return PI*radius*radius;       | (defun areaOfCircle (radius)
                                          (return (* PI radius radius))) |
| r = 10;               | r = 10          | (setq r 10)              |
| area = areaOfCircle(r); | area = areaOfCircle(r) | (setq area (areaOfCircle r)) |
### Programming Languages

<table>
<thead>
<tr>
<th>Processing</th>
<th>Python</th>
<th>Lisp</th>
</tr>
</thead>
</table>
| int areaOfCircle(int radius){
   return PI*radius*radius;
}
| def areaOfCircle(radius):
   return PI*radius*radius;
| (defun areaOfCircle (radius) (return (* PI radius radius))) |
| r = 10;
area = areaOfCircle(r); |
| r = 10
area = areaOfCircle(r) |
| (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!

---

### A Processing Program to Draw a Cat

```java
void draw() {
  // Body
  strokeWeight(5);
  ellipse(200, 130, 160, 120);
  // Face
  strokeWeight(5);
  ellipse(100, 100, 100, 100);
  // Ears
  // left ear
  strokeWeight(5);
  fill(0);
  beginShape();
  vertex(55, 80);
  vertex(50, 50);
  vertex(80, 55);
  endShape(CLOSE);
  // right ear
  strokeWeight(5);
  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, 0, PI/2);
  //ellipse(220, 170, 20, 20, 0, PI/2);
  // Legs and feet
  // left
  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
  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()
```
How to draw an owl...

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

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

    // Nose/Beak
    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()
```

---

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!
Creative
Introduction to Computing

Let’s get started...
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.
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)
Anatomy of a Function Call

Function name  Parentheses

```
line( 10, 10, 50, 80 );
```

Arguments  Statement terminator
Coordinate System

Pixels

*Image of a tree with a circle zoomed into it.*
Processing Canvas

\textbf{size( width, height );}

Set the size of the canvas.

\textbf{background( [0..255] );}

Set the background grayscale color.

Drawing Primitives

\textbf{point( x, y );}

\textbf{line( x1, y1, x2, y2 );}

\textbf{triangle( x1, y1, x2, y2, x3, y3 );}

\textbf{quad( x1, y1, x2, y2, x3, y3, x4, y4 );}

\textbf{rect( x, y width, height );}

\textbf{ellipse( x, y width, height );}
Colors

Composed of four elements:

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

Why 0 .. 255?
Shape Formatting

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

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

---

Fill Color

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

noFill();
```
Stroke (Line) Color

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

noStroke();
```

---

**strokeCap()**

```javascript
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()**

```javascript
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
ellipseMode

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

rectMode

```java
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
256 Shades of Gray