



Exam information

- You must attend your own section.
- Covers lectures 1-5.
- You are allowed to reference your own notes and the slides and examples from lecture and lab.
- You will be given a short p5.js reference on the exam.
- Each question is worth 3 points. There are 5 questions.
- You will begin with some starter code that you will download as a zip file. Download this only once!
- Do not change any code already given to you.

1



Academic Integrity

- Do not share the exam with other students.
- Do not share any files you download or modify with other students.
- Do not discuss the contents of the exam, in part or in whole, with any other students until all students have taken the exam and the instructor signals that discussion is allowed. (no Piazza)
- Do not receive assistance from any other student, whether in this class or not, except for a TA or instructor.
- You must upload your final work as a zip file to Autolab by the end of your lab period, unless you have an accommodation. Changes to this file after your lab period will result in a 0 for the exam.

2



Academic Integrity

- Any detected violation of these exam rules will result in immediate failure in this class and a report sent to the Office of Community Standards and Integrity (OCSI) for further investigation and possible university sanctions, which can include suspension or expulsion.
- Submitting your final zip file to Autolab indicates that you abide by this academic integrity policy.

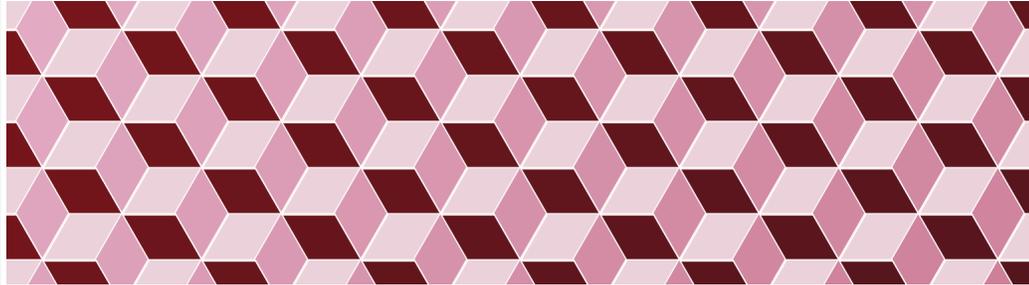
3



Health issues

- If you suspect that you may have been exposed to COVID-19, you must go for a rapid test at University Health Services as soon as possible. You must contact your Housefellow and me to report this issue.
- If this happens, you should be prepared to take the exam remotely under my direction, at the designated time, unless you are admitted for care at a health facility (e.g. hospital). All academic integrity rules still apply.

4



15-104 Introduction to Computing for Creative Practice

Fall 2021

08 Loops

Instructor: Tom Cortina, tcortina@cs.cmu.edu, GHC 4117, 412-268-3514

5



Draw Lots of Lines (source: Shiffman)

```
function setup() {
  createCanvas(480, 270);
  background(200);
}
function draw() {
  stroke(0);
  line(50,60,50,80);
  line(60,60,60,80);
  line(70,60,70,80);
  line(80,60,80,80);
  line(90,60,90,80);
  line(100,60,100,80);
  line(110,60,110,80);
  line(120,60,120,80);
  line(130,60,130,80);
  line(140,60,140,80);
  line(150,60,150,80);
  line(160,60,160,80);
  line(170,60,170,80);
  line(180,60,180,80);
  line(190,60,190,80);
  line(200,60,200,80);
}
```

What patterns can you see here?

6



Finding Patterns

```

var y = 60; // Vertical location of each line
var x = 50; // Initial horizontal location for first line
var spacing = 10; // How far apart is each line
var len = 20; // Length of each line

line(x, y, x, y + len);
// Add spacing so the next line appears 10 pixels to the right
x = x + spacing;

// Continue this process for each line, repeating over and over
line(x, y, x, y + len);
x = x + spacing;

...

```

7



Much Shorter Program with a Loop

```

function setup() {
  createCanvas(480, 270);
  background(200);
}
function draw() {
  var y = 60;
  var spacing = 10;
  var len = 20;
  for (var x = 50; x <= 200; x += spacing) {
    line(x, y, x, y + len); // body of the loop
  }
}

```

Equivalent to:

```

line(50,60,50,80);
line(60,60,60,80);
line(70,60,70,80);
...
line(190,60,190,80);
line(200,60,190,80);

```

8



for Loop

Format:

```
for ( loop_initialization ; loop_condition ; loop_update ) {
    code to repeat
}
```

loop_initialization: defines loop variable that will update with each iteration

loop_condition: defined what condition must be true for the loop to continue to repeat

loop_update: defines any updates that must happen (particularly to the loop variable) at the end of each iteration before the *loop_condition* is tested again.

9



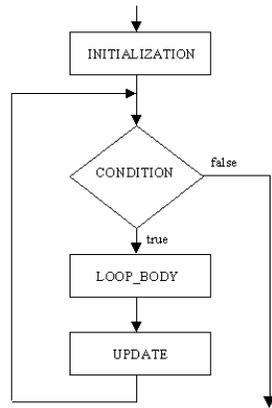
Understanding a loop

```
for (var x = 50; x <= 200; x += spacing) {
    line(x, y, x, y + len);
}
```

1. Set the (local) loop variable **x** equal to 50.
2. If **x** is less than or equal to 200, then run the body of the loop. If not, then exit the loop and continue with the first instruction after the loop.
3. Once one iteration of the body of the loop is completed, update the loop variable by adding **spacing** to it.
4. Go back to step 2.

10

Understanding a loop



Questions:

What is the minimum number of times a loop can repeat?

What is the maximum number of times a loop can repeat?

11

Understanding a loop

Zero iterations:

```
for (var x = 50; x <= 40; x += spacing) {
  line(x, y, x, y + len);
}
```

Infinite Loop:

```
for (var x = 50; x >= 50; x += spacing) {
  line(x, y, x, y + len);
}
```

When using loops, it really helps to print out the loop variable to the console sometimes!

12

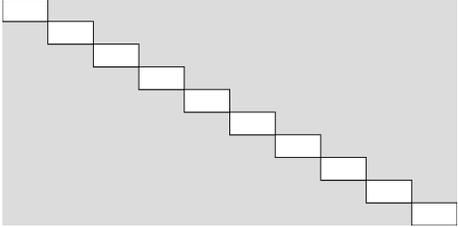


Staircase

```
function setup() {
  createCanvas(500, 250);
  background(220);
}

function draw() {
  var w = width/10;
  var h = height/10;
  var y = 0;

  for (var x = 0; x < width; x += w) {
    rect(x, y, w, h);
    y += h;
  }
}
```



13



Formalities

```
for (var i = 0 ; i < n ; i += 1) {
  // loop body
}
```

This is how programmers typically write a loop that runs n times where i is the loop counter. ($n > 0$)

The variable i cannot be used outside of the loop since it is defined locally (within the loop structure).

The loop counter does not need to be used in the loop body.

Loops can be used within loops!

14