Algorithms and Code

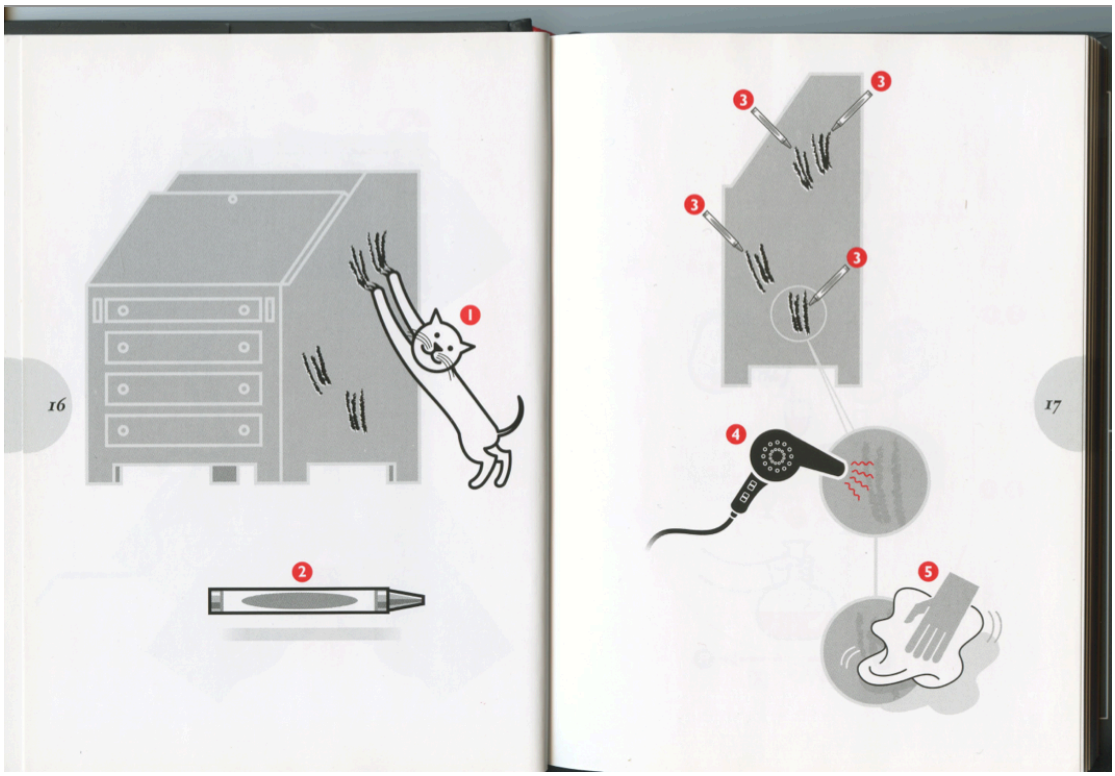
register clicker:
https://www.student.cs.uwaterloo.ca/~cs105/cgi-bin/clicker-form.cgi

activate clicker:
hold ON/OFF, wait for power light to flash, enter room code

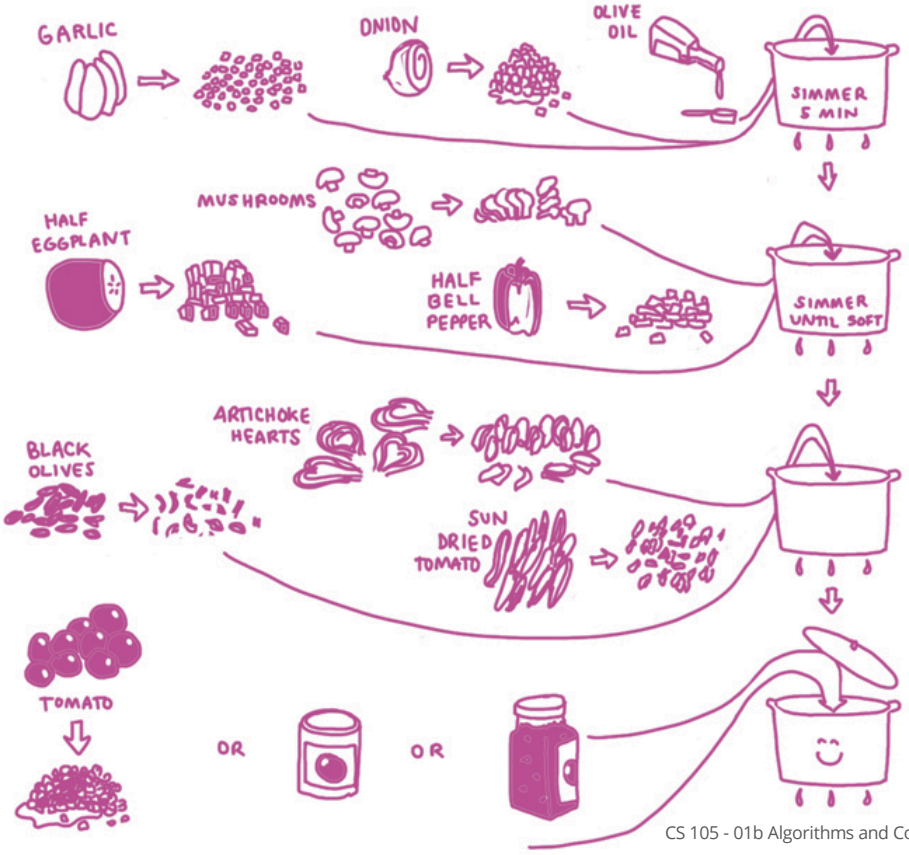
Test your Clicker

register clicker:
https://www.student.cs.uwaterloo.ca/~cs105/cgi-bin/clicker-form.cgi

activate clicker:
hold ON/OFF, wait for power light to flash, enter room code



Credit: Wordless Diagrams



ingredients

- 1 cup olive oil
- 13 cloves garlic
- One 96-ounce can (or, if you can find it, 1-kg) or four 28-ounce cans Italian tomatoes
- Large pinch of red pepper flakes
- 2 teaspoons fine sea salt

preparation

1. Combine the olive oil and garlic in a large deep saucepan and cook over medium-low heat for about 10 minutes, stirring or swirling occasionally, until the garlic is deeply colored—striations of deep brown running through golden cloves—and fragrant. If the garlic starts to smell acrid or sharp or is taking on color quickly, pull the pan off the stove and reduce the heat.

2. While the garlic is getting golden, deal with the tomatoes: Pour them into a bowl and crush them with your hands. We like to pull out the firmer stem end from each of the tomatoes as we crush them and discard those along with the basil leaves that are packed into a can.

3. When the garlic is just about done, add the red pepper flakes to the oil and cook them for 30 seconds or a minute, to infuse their flavor and spice into the oil. Dump in the tomatoes, add the salt, and stir well. Turn the heat up to medium, get the sauce simmering at a gentle pace, not aggressively, and simmer for

Expressing Algorithms

KnitPicks®

LITTLE LEAVES DISHCLOTH

by Jenny Konopinski

(yo and insert hook in st, yo and draw loop through st, yo and draw through first 2 loops on hook) 3 times all into the same st, yo and draw loop through all 4 loops on hook to complete the bobble.

DIRECTIONS

Loosely ch 40 stitches.

Row 1: work 1 bobble in 4th ch from hook, ch 1, *sk 1 ch, 1 bobble in next ch * repeat to end, turn.

Row 2: Ch 3, *1 bobble in next 1-ch sp between bobbles of previous row, ch 1* repeat to end, working last bobble in tch , turn.

Repeat Row 2 until dishcloth measures approx. 10" in length.

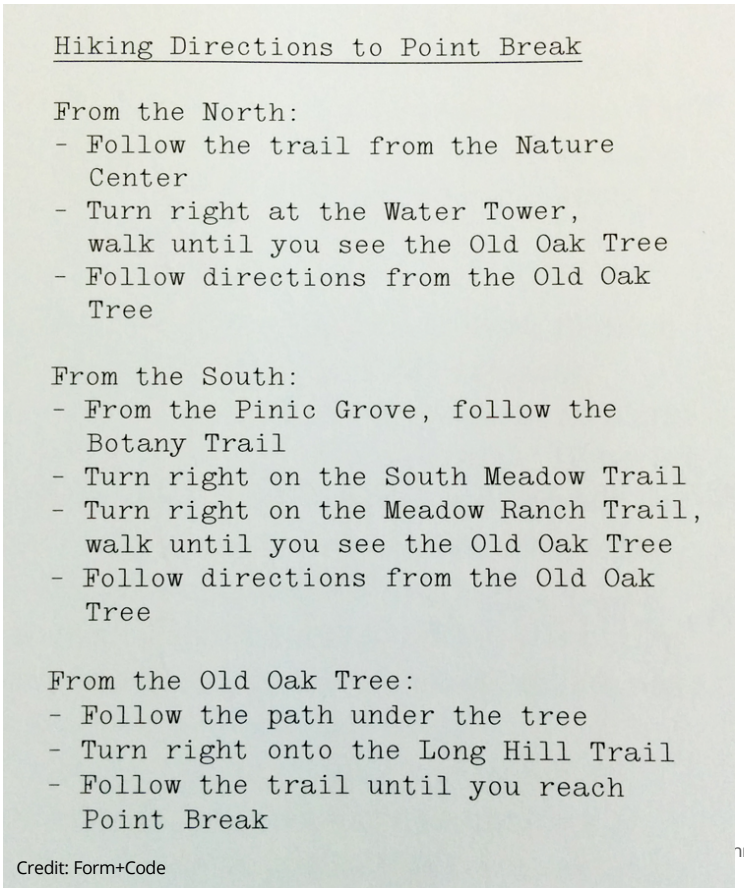
ABBREVIATIONS

ch	chain
sk	skip
sp	space
st	stitch
tch	turning chain
yo	yarn over

yards/50g): Jalapeno 25785 — not critical

HOOKS

http://www.knitpicks.com/patterns/Little_Leaves_Crochet_Dishcloth_-_D55583220.html



Hiking Directions to Point Break

From the North:
- Follow the trail from the Nature Center
- Turn right at the Water Tower, walk until you see the Old Oak Tree
- Follow directions from the Old Oak Tree

From the South:
- From the Pinic Grove, follow the Botany Trail
- Turn right on the South Meadow Trail
- Turn right on the Meadow Ranch Trail, walk until you see the Old Oak Tree
- Follow directions from the Old Oak Tree

From the Old Oak Tree:
- Follow the path under the tree
- Turn right onto the Long Hill Trail
- Follow the trail until you reach Point Break

Credit: Form+Code

Algorithm

An algorithm is a specific set of instructions for carrying out a procedure or for solving a problem.

- It must **produce a result**.
- It must be **achievable/possible**.
- It must be **expressed clearly**.

http://mathworld.wolfram.com/Algorithm.html

More Examples

- What are other examples of algorithms in daily life?

Algorithm Qualities

Algorithms *typically* ...

... make some **assumptions**.

... have **multiple** solutions.

... include **decisions**.

... are expressed in **modular** pieces.

Algorithm Clarity and Precision

Within four adjacent squares,
each 4' by 4',
four draftsmen will be employed
at $4.00/hour
for four hours a day
and for four days to draw straight lines
4 inches long
using four different colored pencils;
9H black, red, yellow and blue.
Each draftsmen will use the same color throughout
the four day period,
working on a different square each day.

Sol LeWitt

Credit: http://journal.mattniebuhr.com/

Algorithm Clarity and Precision

SNOW PIECE

Think that snow is falling.
Think that snow is falling everywhere all the time.
When you talk with a person, think that snow is falling between you and on the person.
Stop conversing when you think the person is covered by snow.

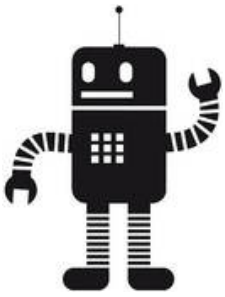
1963 summer

Yoko Ono

credit: http://hi-and-low.typepad.com/

Design an Algorithm

- For controlling a robot



Is it an Algorithm?

Is it an Algorithm?

Computer Algorithm

An algorithm is a well-ordered collection of unambiguous and effectively computable operations that when executed produces a result and halts in a finite amount of time.
[Schneider and Gersting 1995]

1. Algorithms are well-ordered.
2. Algorithms have unambiguous operations.
3. Algorithms have effectively computable operations.
4. Algorithms produce a result.
5. Algorithms finish in a finite amount of time.

Computers aren't really that smart.

- They do very simple things
  - arithmetic
  - follow a sequence of steps
  - make a decision when something is true or false
- They do exactly as they're told.

But they do these things really, really fast and very, very consistently.

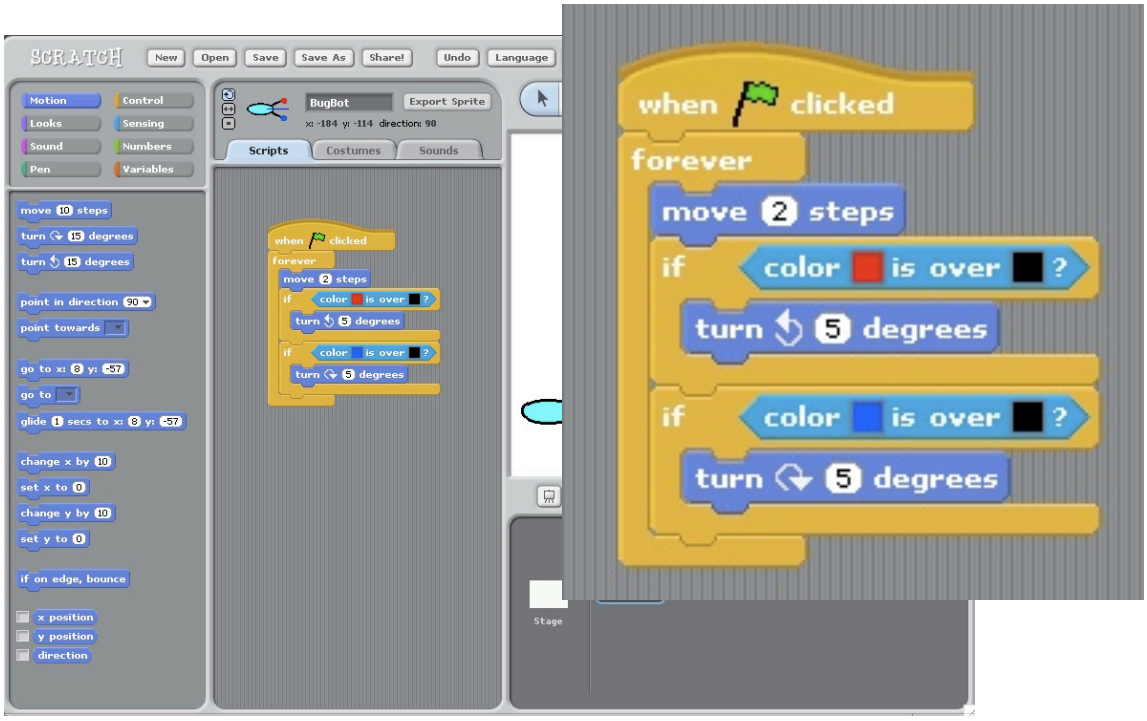
- This can make them appear intelligent: http://nlp-addiction.com/eliza/

Credit: http://mashable.com/2014/08/18/siri-fails/

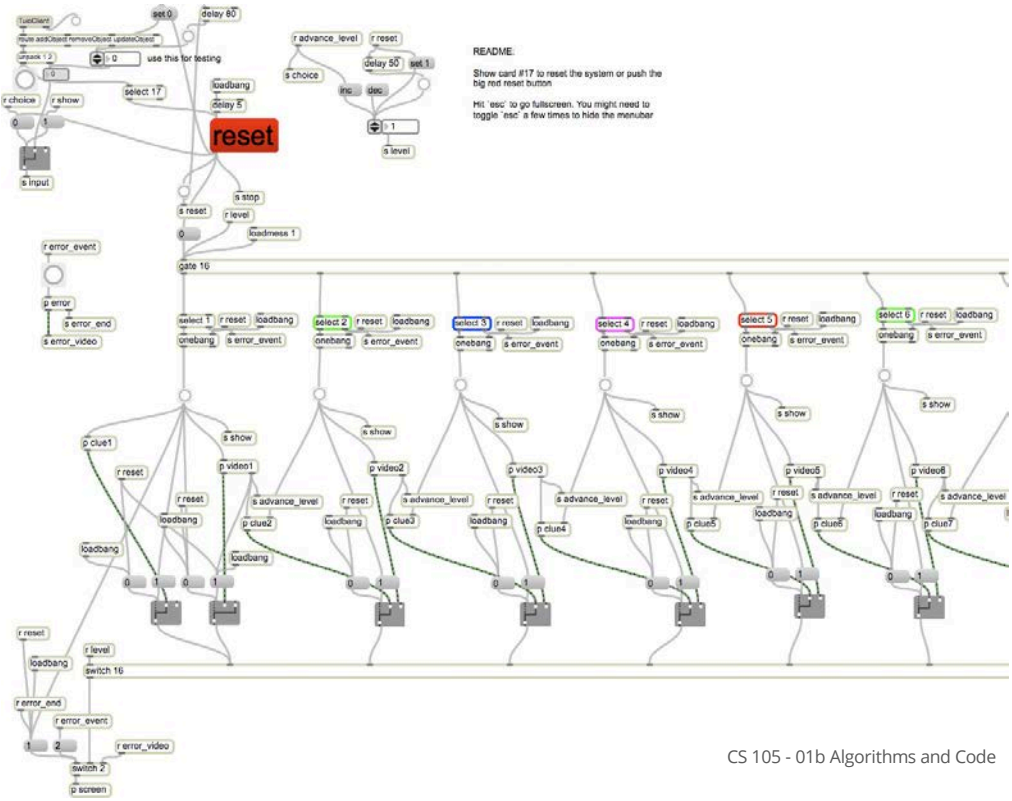
Dials, Knobs, and Lights (1940s)

Howard Aiken, IBM ASCC / Harvard Mark I

Visual Programming Languages (Scratch)



Visual Programming Languages (MAX/MSP)



Code

Codes can be for communication, clarification, obfuscation.

Examples …

- Morse Code
- Health Code
- Secret Code

We focus on code that communicates a set of instructions.

Mark Up Languages (e.g. HTML)

```
    <div id="site" class="page with-navigation clearfix">
  <div id="skip">

  <div id="header" />

<div id="uw-header" class="clearfix">

        <a id="uw-logo" href="//uwaterloo.ca/" accesskey="1">University of Waterloo</a>

        <div id="uw-search">
                <form method="get" action="//uwaterloo.ca/search">
                        <div>
                                <label id="uw-search-label" for="uw-search-term">Search</label>
                                <input id="uw-search-term" type="text" size="31" tabindex="2" accesskey="4"
name="q" />
                                <input id="uw-search-submit" class="chevron-submit" type="submit"
value="Search" tabindex="3" />
                                <input type="hidden" name="client" value="default_frontend" />
                                <input type="hidden" name="proxystylesheet" value="default_frontend" />
                        </div>
                </form>
        </div>


<ul class="global-menu"><li><a href="//uwaterloo.ca/about/">About Waterloo</a></li><li><a
href="//uwaterloo.ca/faculties-academics/">Faculties &amp;amp; Academics</a></li><li><a
href="//uwaterloo.ca/offices-services/">Offices &amp;amp; Services</a></li><li><a
href="http://campaign.uwaterloo.ca/">Support Waterloo</a></li></ul>
</div>
    <div id="site-header"><a href="/stratford-campus/" title="Stratford Campus" rel="home"><img
src="https://uwaterloo.ca/stratford-campus/sites/ca.stratford-campus/files/stratford-campus-wordmark.png"
alt="Stratford Campus"/></a></div>
        <div id="site-navigation">  <div class="region region-sidebar-first">
    <div id="main-menu" class="site-menu">
  <ul class="menu"><li class="first leaf stratford-campus-home mid-490"><a href="/stratford-campus/"
title="">Stratford Campus home</a></li>
<li class="collapsed about-stratford-campus mid-491"><a href="/stratford-campus/about">About Stratford
Campus</a></li>
<li class="expanded active-trail research mid-506"><a href="/stratford-campus/research" class="active-
trail">Research</a><ul class="menu"><li class="first expanded active-trail microtile-wall mid-609"><a
href="/stratford-campus/research/microtile-wall" class="active-trail active">MicroTile Wall</a><ul
class="menu"><li class="first leaf tutorials mid-1305"><a href="/stratford-campus/research/microtile-
wall/tutorials">Tutorials</a></li>
<li class="last leaf after-effects-template mid-1307"><a href="/stratford-campus/research/microtile-
wall/after-effects-template">After Effects template</a></li>
</ul></li>
<li class="last leaf engage-ux-gamification-research-lab mid-1287"><a href="/stratford-campus/research/engage-
```

Machine Code

```
b8    6f 72 6c 64   #moving "orld" into eax
a3    08 10 00 06   #moving eax into next memory location
b8    6f 2c 20 57   #moving ”o wo" into eax
a3    04 10 00 06   #moving eax into next memory location
b8    48 65 6c 6c   #moving ”hell" into eax
a3    00 10 00 06   #moving eax into next memory location
b9    00 10 00 06   #moving pointer to start into ecx
ba    10 00 00 00   #moving string size into edx
bb    01 00 00 00   #moving "stdout" number to ebx
b8    04 00 00 00   #moving "print out" syscall number to eax
cd    80            #calling the kernel to execute print stdout
b8    01 00 00 00   #moving "sys_exit" call number to eax
cd    80            #executing it via sys_call
```

(for Linux, example code is not strictly correct)

Assembly Code

```
; “hello world” program
section .text
global c3Start
c3Start:
 push dword msglen
 push dword mymsg
 push dwod 1
 mov eax, 0x4
 sub esp, 4
 int 0x80
 add esp, 20
 push dword 0
 mov eax, 0x1
 sub esp, 4
 int 0x80
section .data
 mymsg db “hello world”, 0xa
 msglen equ $-mymsg
```

Java

```
// “hello world” program
public class Hello {
      public static void main(String[] args) {
            System.out.println(“hello world”);
      }
}
```


Processing

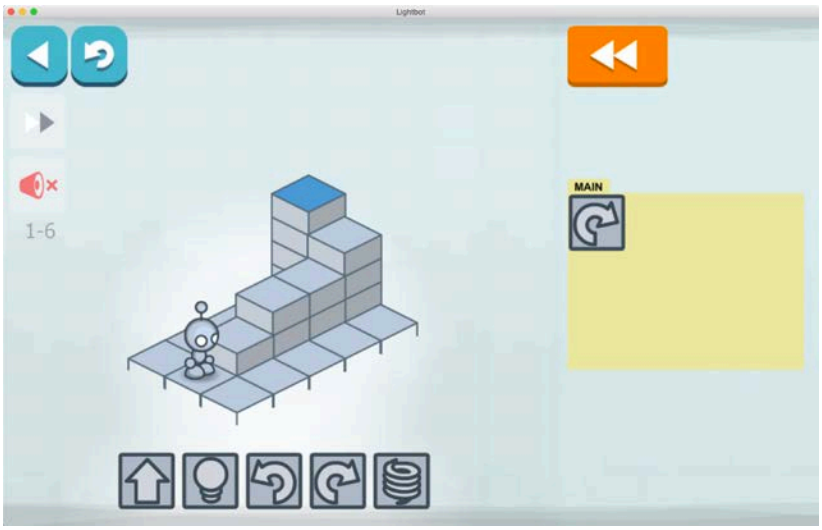
```
// “hello world” program
println(“hello world”);
```

Pseudo-Code

```
print “hello world”
```


Lightbot

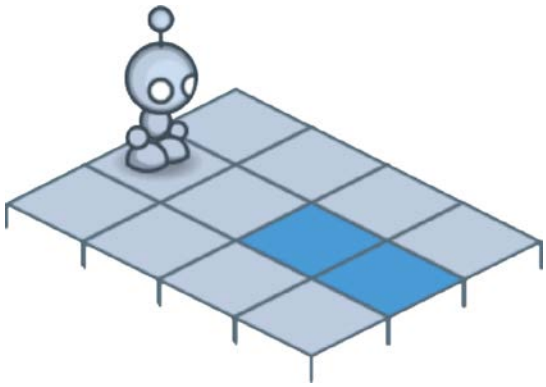
- https://lightbot.com



Lightbot Programming

Language #1

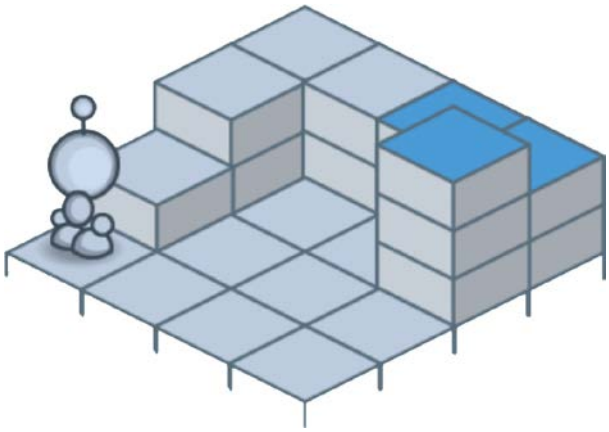
```
forward();
turnRight();
turnLeft();
jump();
light();
```



Lightbot Programming

Language #1

```
forward();
turnRight();
turnLeft();
jump();
light();
```



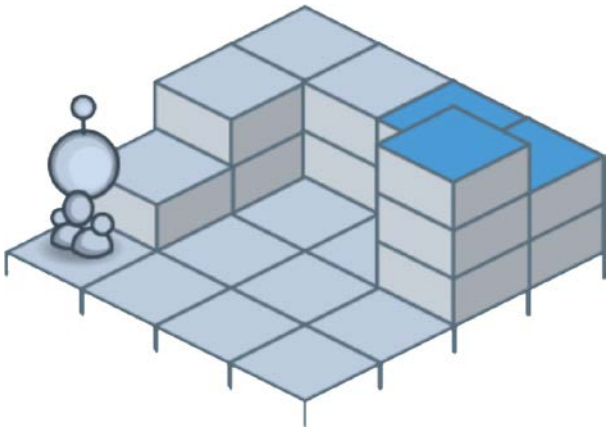
Lightbot Programming

Language #2

```
forward();
turn(X);
```

where X can be LEFT or RIGHT

```
jump();
light();
```



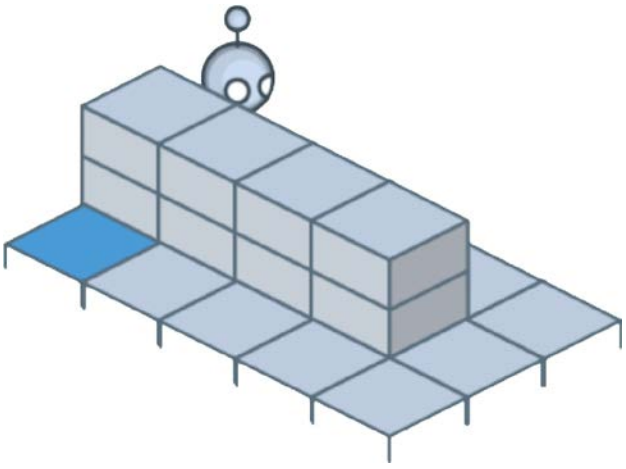
Lightbot Programming

Language #2

```
forward();
turn(X);
```

where X can be LEFT or RIGHT

```
jump();
light();
```



Lightbot Programming

Language #3

`forward(S);`

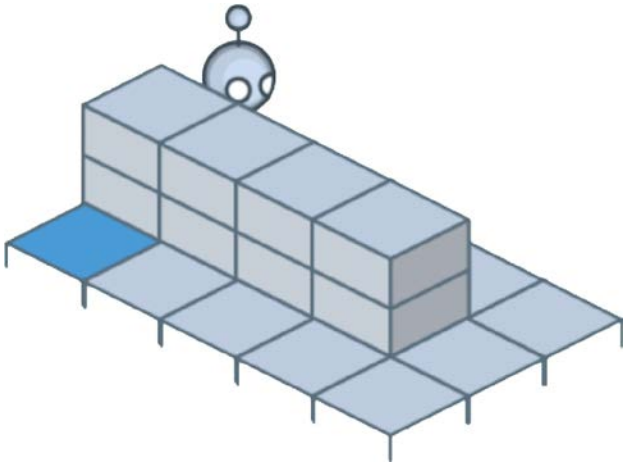
where S is the number of spaces

`turn(X);`

where X can be LEFT or RIGHT

`jump();`

`light();`



Programming Errors

- Syntax Errors
- Runtime Errors
- Logic Errors

Coding Style

- Read introduction and basic in-line spacing in Code Style Guide (on LEARN)

CS 115X Introduction to Computer Science 1

Code Style Guide

Last revised: September 14, 2015

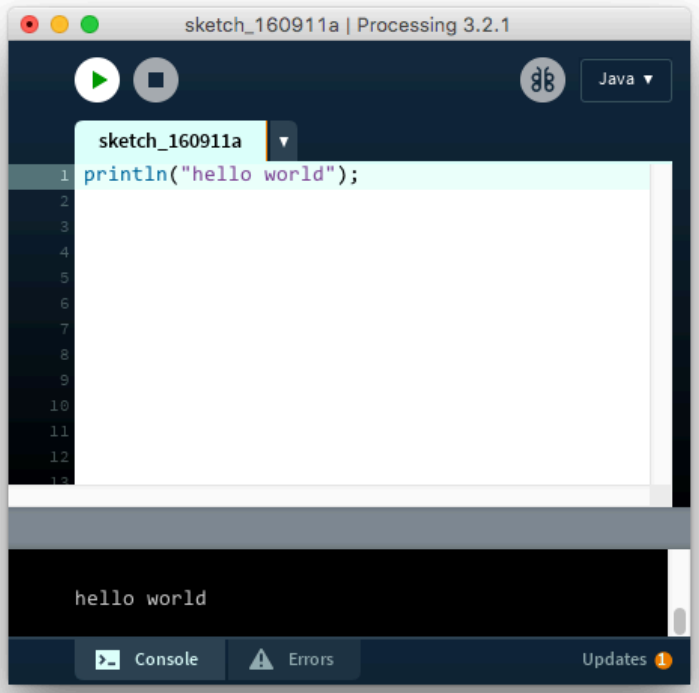
Introduction

The code you submit for labs and assignments is made more readable by paying attention to style. This includes use of whitespace, placement of comments, and choice of variable and function names. None of these affect the way the program executes, but they do affect the readability of your code.

Just like English, computer code is a language. The main goal of all languages is to facilitate communication. When writing code, you are not only communicating with a computer, but also with yourself (often your "future" self) and other people reading your code. In industry,

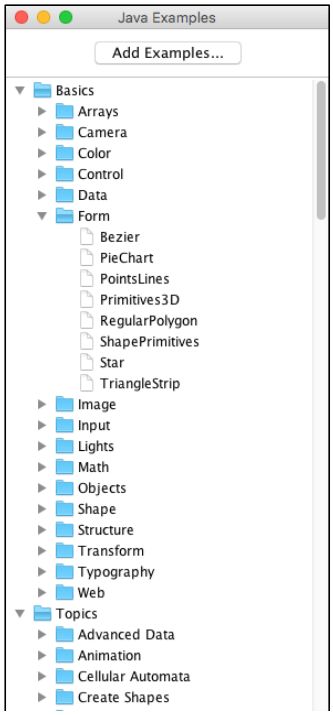
Is this a syntax error in Lightbot Programming Language #1?

Processing IDE



Examples and Reference

File/Examples...



http://www.processing.org/reference/

Reference. The Processing Language was designed to facilitate the creation of sophisticated visual structures.

Structure

() (parentheses)
, (comma)
. (dot)
/* */ (multiline comment)
/** */ (doc comment)
// (comment)
; (semicolon)
= (assign)
[] (array access)
{} (curly braces)
catch
class
draw()
exit()
extends
false
final
implements

Shape

createShape()
loadShape()
PShape

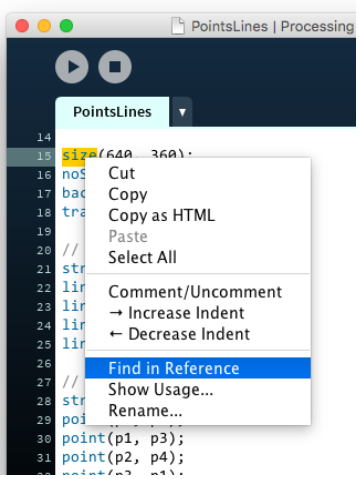
2D Primitives
arc()
ellipse()
line()
point()
quad()
rect()
triangle()

Curves
bezier()
bezierDetail()
bezierPoint()
bezierTangent()

Color

Setting
background()
clear()
colorMode()

Control-Click,
Find in Reference



CS

Online Processing Resources

- Learning
  - http://www.processing.org/tutorials/
  - http://processing.org/examples/
  - http://www.codecademy.com/
  - http://www.learningprocessing.com/examples/

- Inspiration
  - http://processingjs.org/exhibition/
  - http://www.openprocessing.org/