# About CS 101, Fall 2019

Chris Mayfield, Sharon Simmons, Michael Stewart
Department of Computer Science
James Madison University

(click here for video)

## Course design

**survey course**

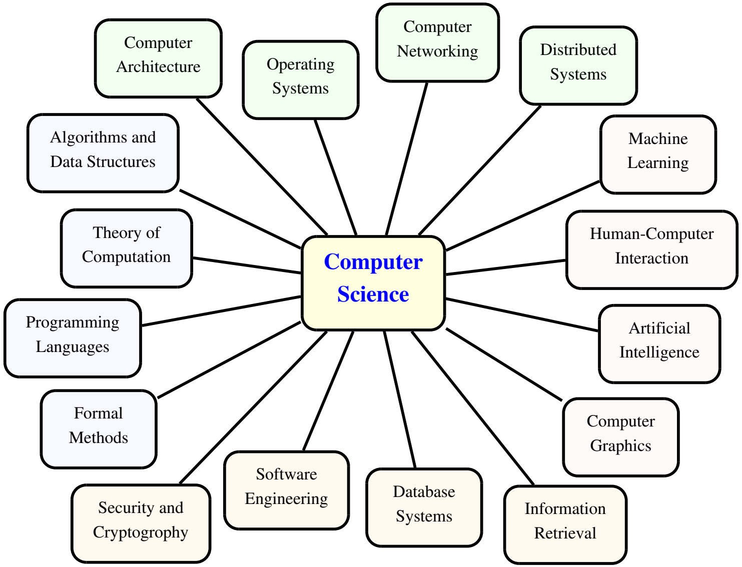
*n.*

> *An academic course consisting of an overview of a broad topic or field of knowledge.*
>
> *(American Heritage Dictionary)*

What this means:

- You will learn about many topics
- Focus on breadth, not on depth
- See the big picture of what is CS

## Example sub-fields



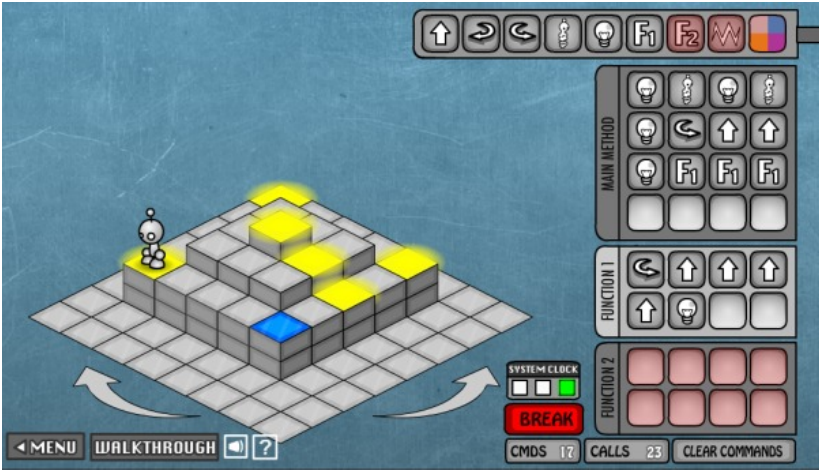
## Overall theme

How to think like a computer scientist

## Example Learning Objective

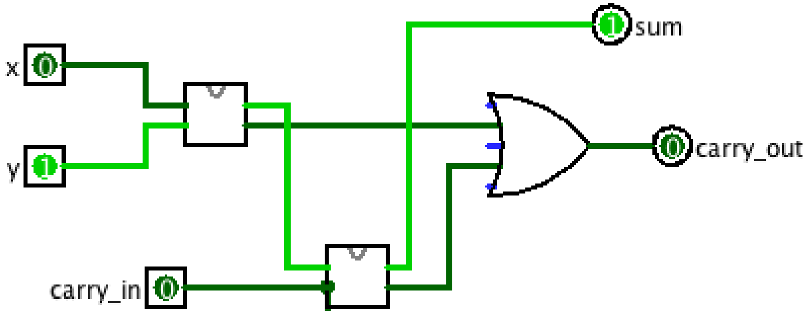
Use abstraction and decomposition when reasoning about complex systems and problems.

## Lab01 example



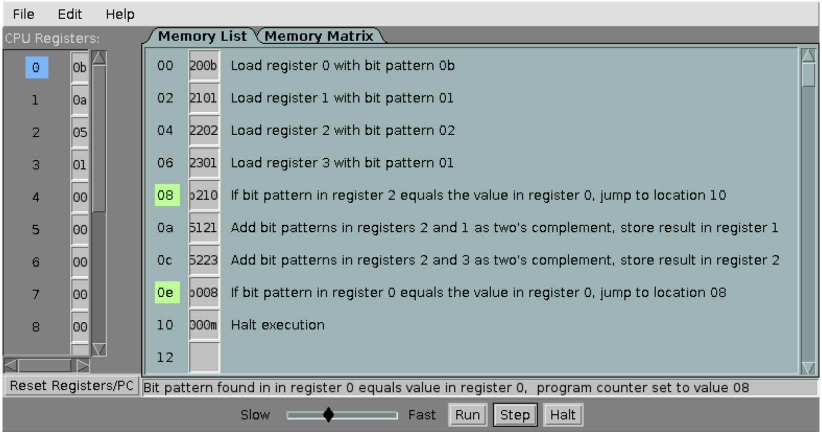
"F1" is an abstract tool

## Lab02 example



“Half adder” is an abstract tool

# Lab03 example



| | |
|---|---|
| 0 | 0b |
| 1 | 0a |
| 2 | 05 |
| 3 | 01 |
| 4 | 00 |
| 5 | 00 |
| 6 | 00 |
| 7 | 00 |
| 8 | 00 |

| | | |
|---|---|---|
| 00 | 200b | Load register 0 with bit pattern 0b |
| 02 | 2101 | Load register 1 with bit pattern 01 |
| 04 | 2202 | Load register 2 with bit pattern 02 |
| 06 | 2301 | Load register 3 with bit pattern 01 |
| 08 | b210 | If bit pattern in register 2 equals the value in register 0, jump to location 10 |
| 0a | 5121 | Add bit patterns in registers 2 and 1 as two's complement, store result in register 1 |
| 0c | 5223 | Add bit patterns in registers 2 and 3 as two's complement, store result in register 2 |
| 0e | b008 | If bit pattern in register 0 equals the value in register 0, jump to location 08 |
| 10 | 000m | Halt execution |
| 12 | | |

Bit pattern found in in register 0 equals value in register 0,  program counter set to value 08

"CPU instructions" are abstract tools

## Essence of CS 101

The point is NOT:

- Become an expert at Light-Bot programming
- Be able to understand/design complex circuits
- Program a computer in machine language

The point is:

- Learn how to think like a computer scientist
- Sample what you will learn in future courses
- Develop new computing skills (e.g., Python)

## Textbook

Computer Science: An Overview

Brookshear and Brylow, 12th edition

http://www.pearsonhighered.com/brookshear

## Course outline

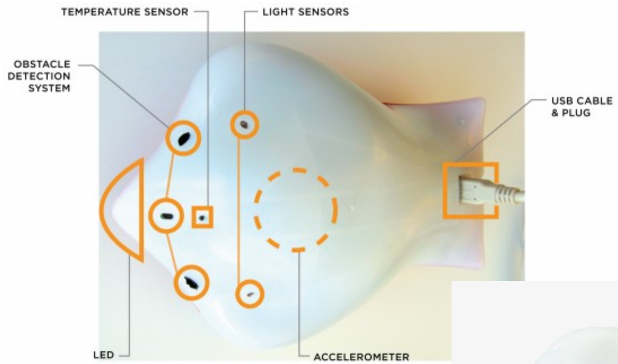
### Part 1: Hardware and Systems

1. Introduction
2. Data Storage
3. Program Execution
4. Operating Systems
5. Computer Networking
6. Information Security
7. Mid Project: Explore

### Part 2: Software and Data

8. Algorithms and Python
9. Programming Languages
10. Software Engineering
11. Data Structures
12. Database Systems
13. Artificial Intelligence
14. Final Project: Create

# Finch robots!



## Big Ideas of Computer Science

Source: http://apcsprinciples.org/

(see also Section 0.4 of the book)

## Big Idea 1: Creativity

Computing is a creative activity.

## Big Idea 2: Abstraction

Abstraction reduces information and detail to facilitate focus on relevant concepts.

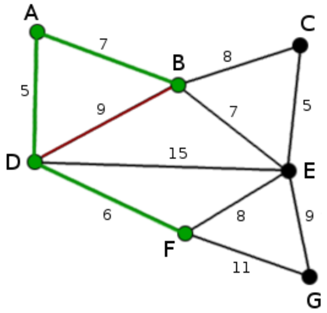
## Big Idea 3: Data and Information

http://www.ibm.com/big-data

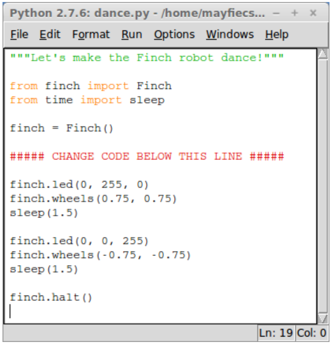
Data and information facilitate the creation of knowledge.

# Big Idea 4: Algorithms

Algorithms are used to develop and express solutions to computational problems.



## Big Idea 5: Programming

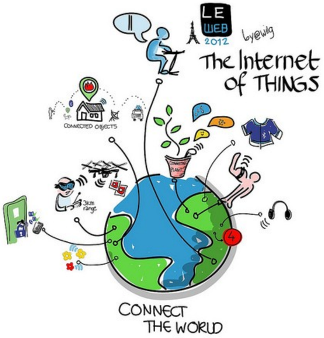

Python 2.7.6: dance.py - /home/mayfiecs...
File Edit Format Run Options Windows Help
Ln: 19 Col: 0

```
"""Let's make the Finch robot dance!"""

from finch import Finch
from time import sleep

finch = Finch()

##### CHANGE CODE BELOW THIS LINE #####

finch.led(0, 255, 0)
finch.wheels(0.75, 0.75)
sleep(1.5)

finch.led(0, 0, 255)
finch.wheels(-0.75, -0.75)
sleep(1.5)

finch.halt()
```

Programming enables problem solving, human expression, and creation of knowledge.

## Big Idea 6: The Internet

The Internet pervades modern computing.



*by Wilgengebroed on Flickr*

## Big Idea 7: Global Impact

*Source: smallbiztrends.com*

Computing has global impact.