Quest (first exam) in **in 14 days!!**

UC Berkeley EECS
Lecturer SOE
Dan Garcia

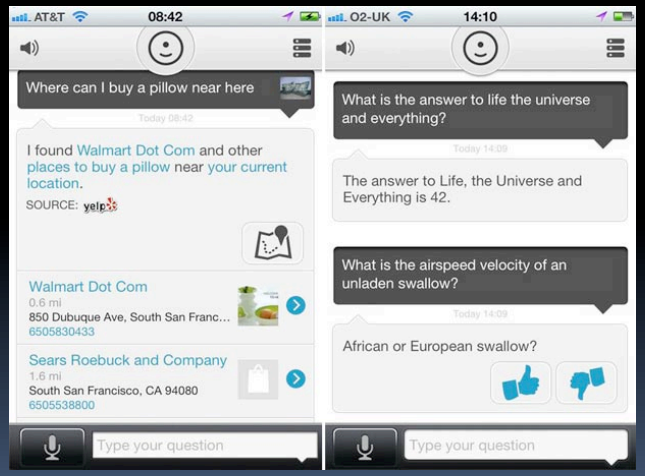
CS10 : The Beauty and Joy of Computing

Lecture #4
Functions

2012-01-30

SIRI COMPETITION? EVI

The success of Apple's Siri (only available on the iPhone 4S) has sparked competition, to be sure. Google's IRIS (Siri spelled backward), and now Evi (available on BOTH iOS and Android). The popularity has meant the servers are down (they didn't use Cloud storage clearly – we'll learn about that later). Love where this is going!



www.technologyreview.com/computing/39560/

Enrollment – everyone IS in

Course:	COMPUTER SCIENCE 10 P 001 LEC
Course Title:	The Beauty and Joy of Computing (catalog description)
Location:	MW 3-4P, 145 DWINELLE
Instructor:	GARCIA, D D
Status/Last Changed:	UPDATED: 07/27/11
Course Control Number:	26230 View Books
Units/Credit:	4
Final Exam Group:	8: TUESDAY, DECEMBER 13, 2011 7-10P
Restrictions:	UG
Note:	
Enrollment on 09/09/11:	Limit:239 Enrolled: 8 Waitlist:0 Av 1 Seats:1

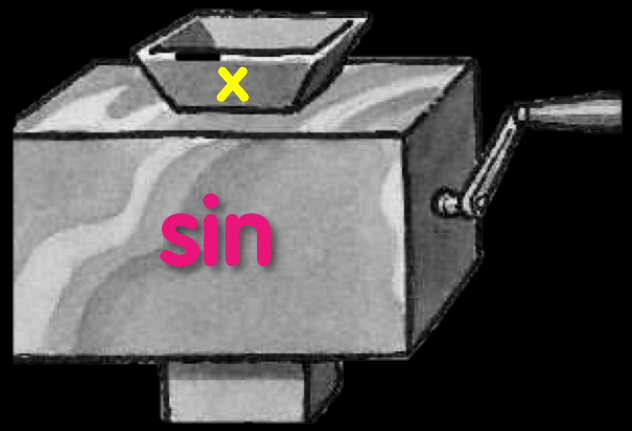
Click here for current enrollment inform nd course restric

Generalization (in CS10) REVIEW

- You are going to learn to write functions, like in math class:

$$y = \sin(x)$$

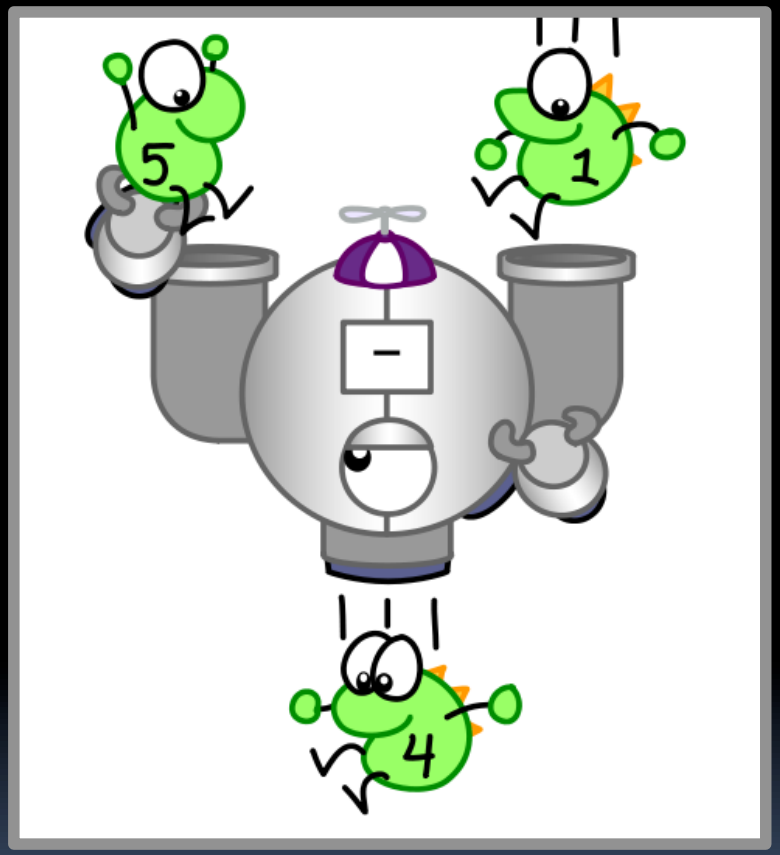
  - sin is the function
  - x is the input
  - It returns a single value, a number



"Function machine" from *Simply Scheme* (Harvey)

Function basics

- Functions take in **0 or more inputs** and return **exactly 1 output**
- The same inputs MUST yield same outputs.
  - Output function of input only
- Other rules of functions
  - No state (prior history)
  - No mutation (no variables get modified)
  - No side effects (nothing else happens)



CS Illustrated function metaphor

Which is NOT a function?

a) pick random ( ) to ( )

b) ( ) < ( )

c) length of ( )

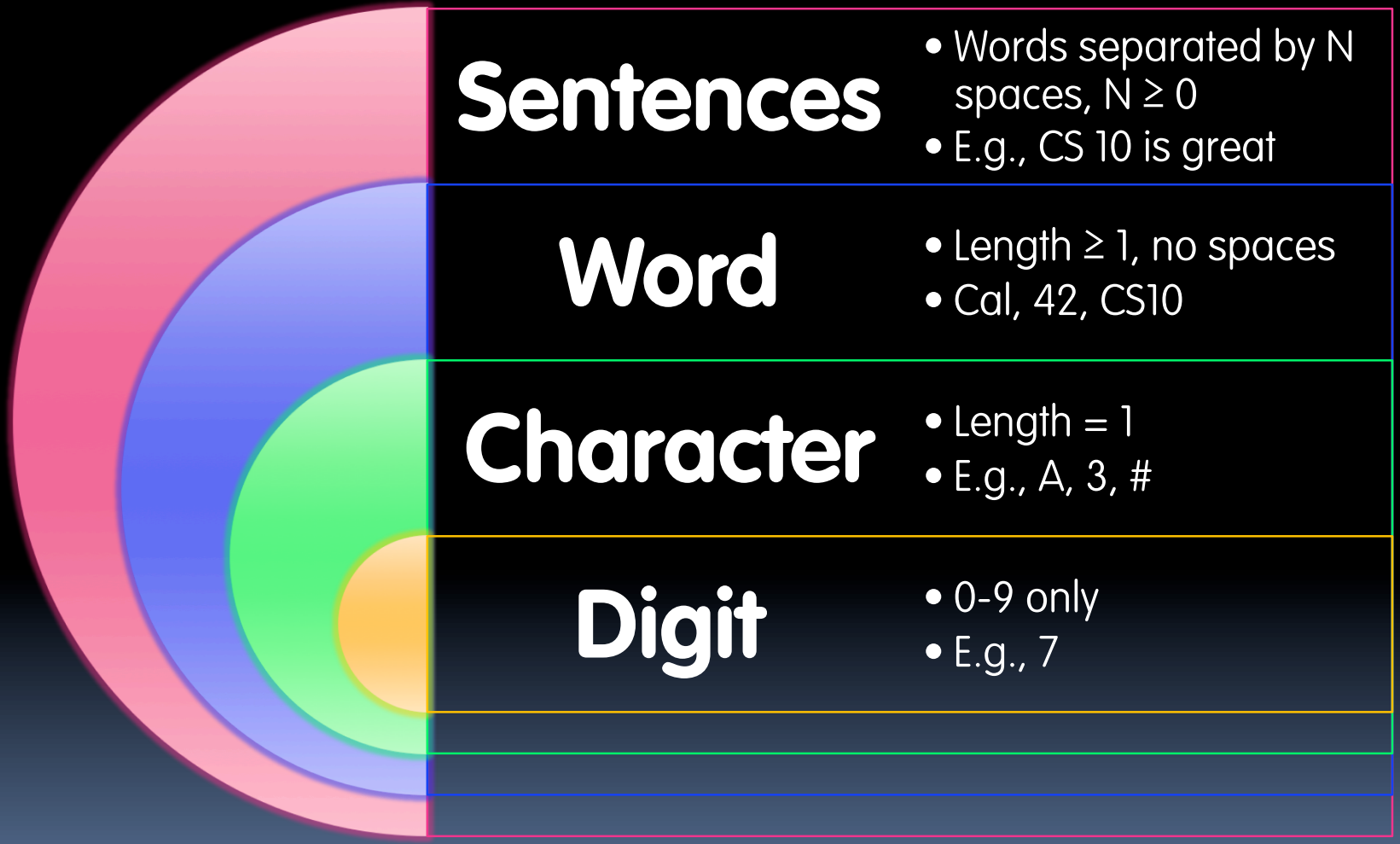
d) sqrt of ( )

e) true

More Terminology (from Math)

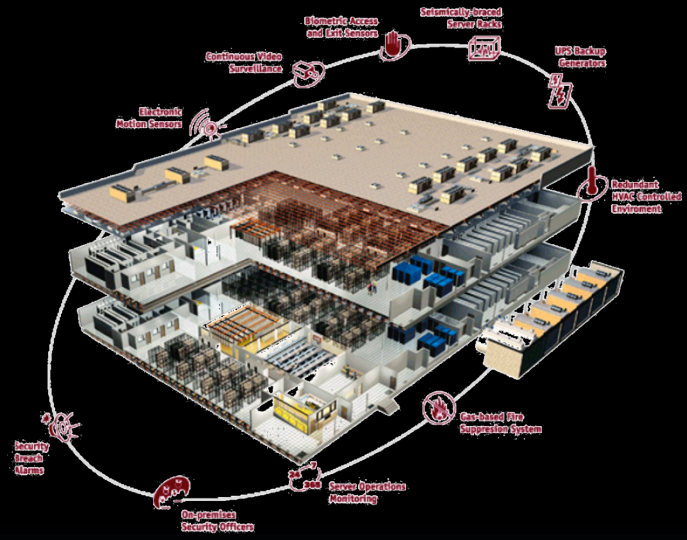
- **Domain**
  - The "class" of input a function accepts
- **Examples**
  - Sqrt of
    - Positive numbers
  - Length of
    - Sentence, word, number
  - _ < _
    - Both: Sentence, word, number
  - _ and _
    - Booleans
  - Letter _ of _
    - Number from 1 to input length
    - Sentence, word, number
- **Range**
  - All the possible return values of a function
- **Examples**
  - Sqrt of
    - Non-negative numbers
  - Length of
    - Non-negative integer
  - _ < _
    - Boolean (true or false)
  - _ and _
    - Boolean (true or false)
  - Letter _ of _
    - Letter

Types of input (there are more)



Why functions are great!

- **If a function only depends on the information it gets as input, then nothing else can affect the output.**
  - It can run on any computer and get the same answer.
- **This makes it incredibly easy to parallelize functions.**
  - Functional programming is a great model for writing software that runs on multiple systems at the same time.



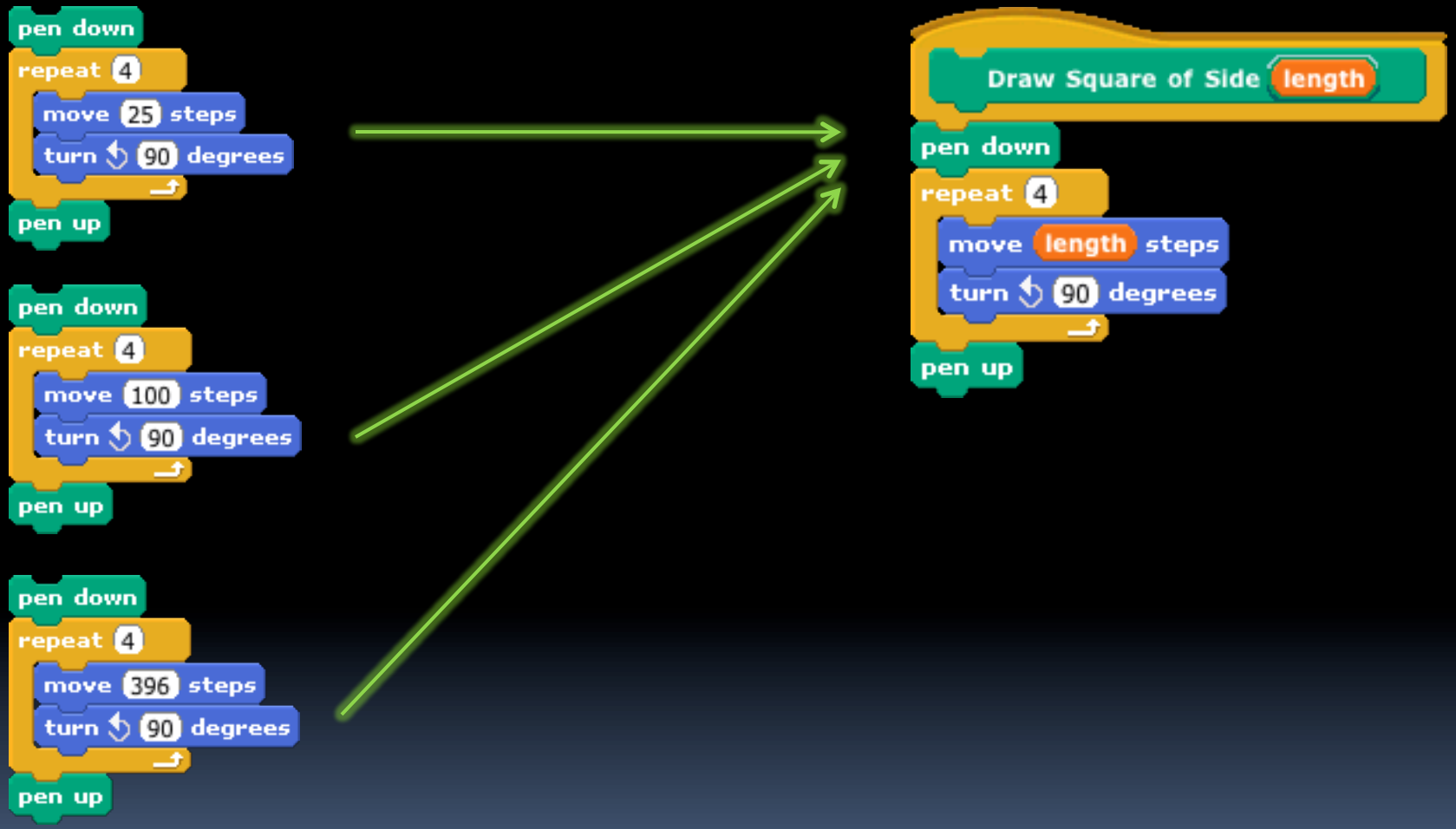
Datacenter

Scratch → BYOB (Build Your Own Blocks)

λSnap!

- **Scratch**
  - Invented @ MIT
  - Maintained by MIT
  - Huge community
  - Sharing via Website
  - No functions ☹
  - Scratch 2.0 in Flash
    - No iOS devices. ☹
  - scratch.mit.edu
- **BYOB (to be "Snap!")**
  - Based on Scratch code
  - Maintained by jens & Cal
  - Growing community
  - No sharing (yet) ☹
  - Functions! ☺ … "Blocks"
  - BYOB 4.0 in HTML5
    - All devices ☺
  - byob.berkeley.edu

Why use functions? (1)



The power of **generalization**!

Why use functions? (2)

They can be composed together to make even more magnificent things.

They are literally the building blocks of almost everything that we create when we program.

We call the process of breaking big problems down into smaller tasks functional decomposition

Types of Blocks

- **Command**
  - No outputs, meant for side-effects

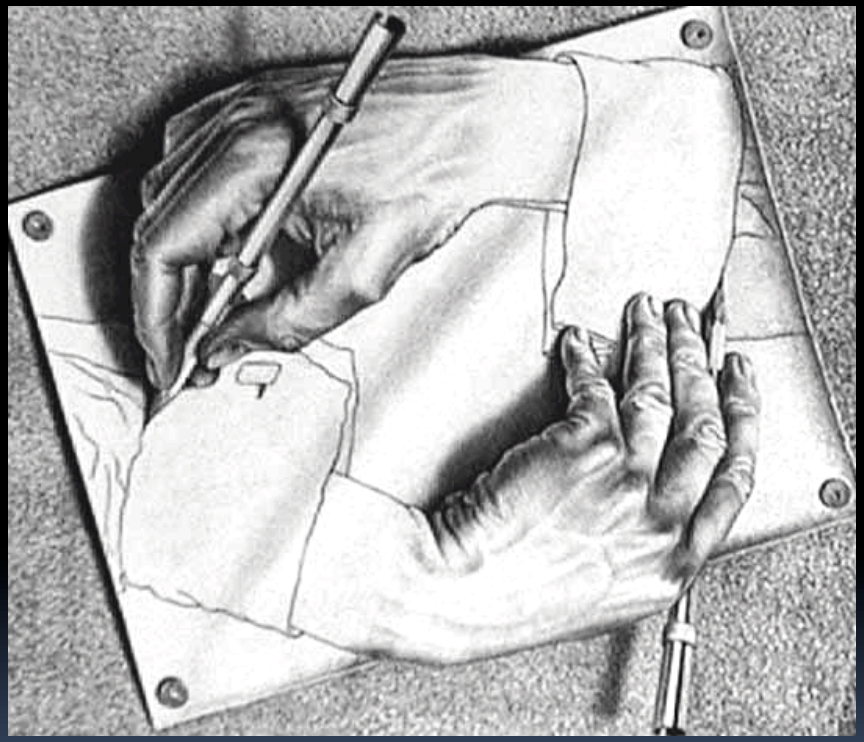
- **Reporter (Function)**
  - Any type of output

- **Predicate (Function)**
  - Boolean output
    - (true or false)

Quick Preview: Recursion

Recursion is a technique for defining functions that use **themselves** to complete their own definition.

M. C. Escher : *Drawing Hands*



We will spend a lot of time on this.

Functional Programming Summary

- **Computation is the evaluation of functions**
  - Plugging pipes together
  - Each pipe, or function, has exactly 1 output
  - Functions can be input!
- **Features**
  - No state
    - E.g., variable assignments
  - No mutation
    - E.g., changing variable values
  - No side effects
- **Need BYOB not Scratch**

$$f(x) = (x+3) * \sqrt{x}$$