


# The Beauty and Joy of Computing


## Lecture #3 : Functions

Instructor: Sean Morris



### DATA YOU CAN BELIEVE IN?

The role of data mining in the last presidential election. A mind-blowing piece on how the Obama campaign used various sources of data to target voters at an intensity unheard of before. Privacy ramifications? [8](#) #cs10news



<http://www.nytimes.com/2013/06/23/magazine/the-obama-campaigns-digital-masterminds-cash-in.html?pagewanted=all>

## BJC in one slide

- Big Ideas of Programming
  - Abstraction
  - Algorithms (2)
  - Recursion (2)
  - Functions-as-data,  $\lambda$  (2)
  - Programming Paradigms
  - Concurrency
  - Distributed Computing
- Beauty and Joy
  - "CS Unplugged" activities
  - Lab work in pairs
  - Two projects in pairs
    - Of your own choice!!
  - One blog
    - Of students' own choice!!
- Big Ideas of Computing
  - HowStuffWorks
    - 3D Graphics
    - Video Games
    - Computational Game Theory
  - Research Summaries
    - AI
    - HCI
  - Apps that Changed the World
  - Social Implications of Computing
  - Saving the World with Computing
  - How Twitter Works
  - Cloud Computing
  - Future of Computing

Garcia  
UC Berkeley "The Beauty and Joy of Computing": Functions (2)

## Today

- Define Functions ( Connected to Abstraction?)
  - Inputs and Output (Domain/Range)
- Why Functions
  - Practical Example
  - The nuts and bolts of generalization
  - A method for solving large problems
- See and work with BYOB


Garcia  
UC Berkeley "The Beauty and Joy of Computing": Functions (3)

## 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)

Garcia  
UC Berkeley "The Beauty and Joy of Computing": Functions (4)

## Sean's kid's 2<sup>nd</sup> grade HW!

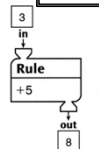
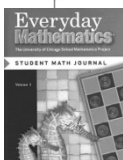
HOME LINK 2-11 "What's My Rule?"

**Family Note** Today your child learned about a kind of problem you may not have seen before. We call it "What's My Rule?" Please ask your child to explain it to you.

Here is a little background information: Imagine a machine with a funnel at the top and a tube coming out of the bottom. The machine can be programmed so that if a number is dropped into the funnel, the machine does something to the number, and a new number comes out of the tube. For example, the machine could be programmed to add 5 to any number that is dropped in. If you put in 3, 8 would come out. If you put in 7, 12 would come out.

We call this device a **function machine**. You can show the results of the rule " $+5$ " in a table:

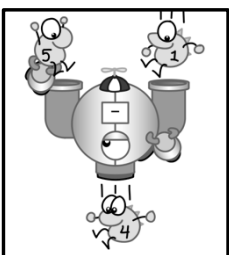
in	out
3	8
7	12
15	20

Garcia  
UC Berkeley "The Beauty and Joy of Computing": Functions (5)

## 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

Garcia  
UC Berkeley "The Beauty and Joy of Computing": Functions (6)

### Which is NOT a function?

- a) `pick random` to
- b) `<`
- c) `length of`
- d) `sqrt of`
- e) `true`



UC Berkeley "The Beauty and Joy of Computing": Functions (7)

### Pair Action...

- Function named: **Calculate Potential Popularity of Movie**
- Input(s)
- Output
- What action might break one of our rules for functions?

UC Berkeley "The Beauty and Joy of Computing": Functions (8)

### More Terminology (from Math)

- Domain**
  - The "class" of input a function accepts
- Range**
  - All the possible return values of a function
- 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

UC Berkeley "The Beauty and Joy of Computing": Functions (9)

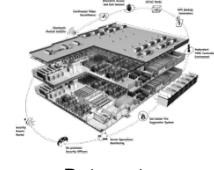
### Types of input (there are more)

<b>Sentences</b>	<ul style="list-style-type: none"> <li>• Words separated by N spaces, <math>N \geq 0</math></li> <li>• E.g., CS 10 is great</li> </ul>
<b>Word</b>	<ul style="list-style-type: none"> <li>• Length <math>\geq 1</math>, no spaces</li> <li>• E.g., Cal, 42, CS10</li> </ul>
<b>Character</b>	<ul style="list-style-type: none"> <li>• Length = 1</li> <li>• E.g., A, 3, #</li> </ul>
<b>Digit</b>	<ul style="list-style-type: none"> <li>• 0-9 only</li> <li>• E.g., 7</li> </ul>

UC Berkeley "The Beauty and Joy of Computing": Functions (10)

### 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


UC Berkeley "The Beauty and Joy of Computing": Functions (11)

### Scratch → BYOB (Build Your Own Blocks)



Scratch

- Invented @ MIT
- Maintained by MIT
- Huge community
- Sharing via Website
- No functions ☹️
- Scratch 2.0 in Flash
  - No iOS devices. ☹️
- [scratch.mit.edu](http://scratch.mit.edu)



BYOB (and Snap!)

- Based on Scratch code
- Maintained by Jens & Cal
- Growing community
- No sharing (yet) ☹️
- Functions! ☺️ ... "Blocks"
- Snap! Is in HTML5
  - All devices ☺️
- [snap.berkeley.edu/run](http://snap.berkeley.edu/run)

UC Berkeley "The Beauty and Joy of Computing": Functions (12)

## Why use functions? (1)

The power of generalization!

UC Berkeley "The Beauty and Joy of Computing": Functions (13)

## 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

UC Berkeley "The Beauty and Joy of Computing": Functions (14)

## Types of Blocks

- Command
  - No outputs, meant for side-effects
  - Not a function...
- Reporter (Function)
  - Any type of output
- Predicate (Function)
  - Boolean output
  - (true or false)

UC Berkeley "The Beauty and Joy of Computing": Functions (15)

## Quick Preview: Recursion

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

We will spend a lot of time on this.

UC Berkeley "The Beauty and Joy of Computing": Functions (16)

[en.wikipedia.org/wiki/Functional\\_programming](https://en.wikipedia.org/wiki/Functional_programming)

## Functions 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/Snap!, and not Scratch 1.x

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

UC Berkeley "The Beauty and Joy of Computing": Functions (17)