

**Computational Structures in Data Science** 



# Lecture #3: **Control Recap & Higher Order Functions**

February 11, 2019

# **Solutions for the Wandering Mind**



Could we build a complete computer that has no instructions, only data?

Yes! A computer that only uses a single instruction doesn't have to distinguish between instructions. The program is a sequence of arguments to that instruction.

One Instruction Computer:

https://en.wikipedia.org/wiki/One instruction set comput

Generalization: Cellular Automaton (Rule F110) https://en.wikipedia.org/wiki/Cellular\_automaton

Is this how the universe works?

UCB CS88 Sp19 L3

# **Administrative issues**



- Tutoring
  - To help you prepare for exams, we will be hosting small group tutoring we will also be having guerrilla section.
  - Pay attention on Piazza and ask TAs for details.
- · Midterm Thursday 3/7. DSP and make-up details

UCB CS88 Sp19 L3

## **Computational Concepts Toolbox**



- · Data type: values, literals, operations, - e.g., int, float, string
- · Expressions, Call expression
- Variables
- · Assignment Statement
- · Sequences: tuple, list
- · Data structures
- · Tuple assignment
- Call Expressions
- · Function Definition Statement · Conditional Statement



- data-driven (list comprehension)
  - control-driven (for statement)
- while statement

UCB CS88 Sp19 L3

#### **Computational Concepts today**



- Recap: Control structuresHigher Order Functions
- · Functions as Values
- · Functions with functions as argument
- · Assignment of function values
- · Higher order function patterns
  - Map. Filter, Reduce
- · Function factories create and return functions



Big Idea: Software Design Patterns

#### for statement - iteration control



Repeat a block of statements for a structured sequence of variable bindings

<initialization statements>

for <variables> in <sequence expression>: <body statements>

<rest of the program>

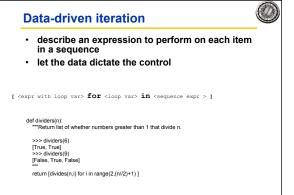
def cum\_OR(lst): """Return cumulative OR of entries in lst.
>>> cum\_OR([True, False])

True >>> cum\_OR([False, False])

for item in lst:

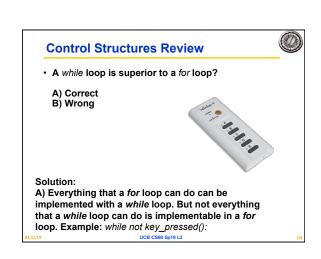
co = co or item return co



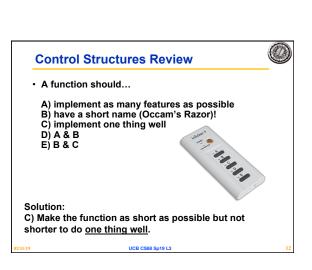


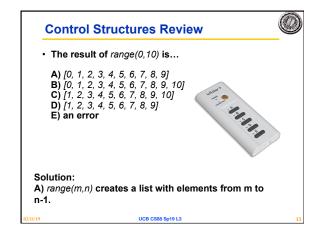
UCB CS88 Sp19 L3

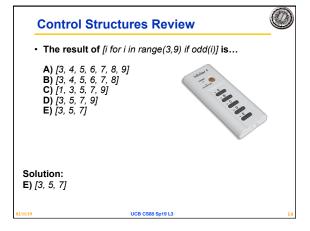


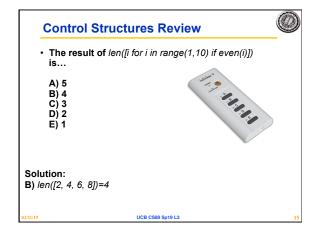














```
Higher Order Functions

• Functions that operate on functions
• A function

def odd(x):
    return (x%2==1)
    >>> odd(3)
    True

• A function that takes a function arg

def filter(fun, s):
    return [x for x in s if fun(x)]
    >>> filter(odd, [0,1,2,3,4,5,6,7])
[1, 3, 5, 7]
```

```
Higher Order Functions (cont)

• A function that returns (makes) a function

def leq_maker(c):
    def leq(val):
        return val <= c
    return leq

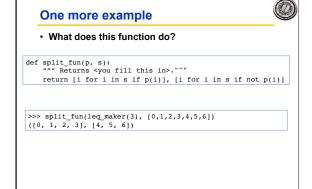
>>> leq_maker(3)
    <function leq_maker.<locals>.leq at 0x1019d8c80>

>>> leq_maker(3)(4)
    False

>>> filter(leq_maker(3), [0,1,2,3,4,5,6,7])
    [0, 1, 2, 3]

>>>

**DULLY OUR CSSS Sp19 L3
```



UCB CS88 Sp19 L3

## **Three super important HOFS**



map(function\_to\_apply, list\_of\_inputs)
Applies function to each element of the list

filter(condition, list\_of\_inputs)
Returns a list of elements for which the
condition is true

reduce(function, list\_of\_inputs)
Reduces the list to a result, given the function

9 UCB CS88 Sp19 L3

#### **Function Factories**



def linemaker(m, b):
 def linefun(x):
# Create a function that embeds the parameters of the line
 return m\*x + b
# Return that dynamically created function
return linefun

def make decoder(code\_map):

"""Make a decoder function specified by a map"""

def decode(code):

for (code\_num, desc) in code\_map:

if code == code\_num:

return desc

return "unknown"

return decode

11/19 UCB CS88 Sp19 L3

# **Computational Concepts today**



- · Higher Order Functions
- · Functions as Values
- Functions with functions as argument
- · Assignment of function values
- Higher order function patterns
   Map, Filter, Reduce
- · Function factories create and return functions



Big Idea: Software Design Patterns

UCB CS88 Sp19 L3

# Thoughts for the Wandering Mind (Holiday Edition)



- How many answers can be maximally responded to by 20 questions (how much data do I need on my game device)?
- · How can a 20-questions game get away with less?
- How can you make a 20-questions game fail (adversarial attack)?



UCB CS88 Sp19 L3