

Review for the Final Exam

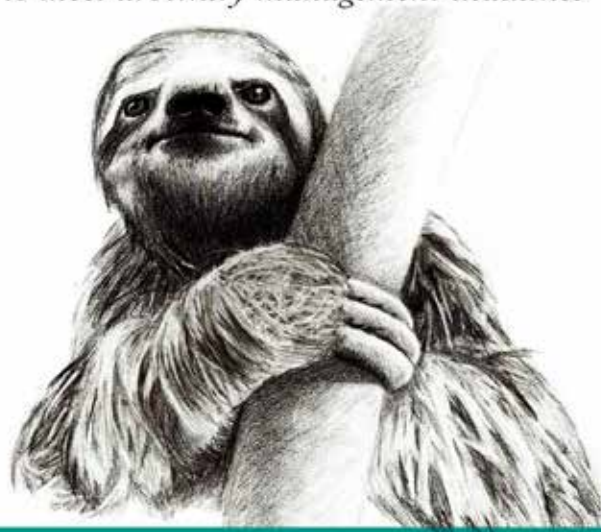
CS 8: Introduction to Computer Science, Winter 2018
Lecture #15

Ziad Matni
Dept. of Computer Science, UCSB

Administrative

- Project #2 is **DUE on FRIDAY – no late submissions accepted**
- Homework #8 due today
- Lab #6 due today
- To collect older homework, come by my office
 - Email ahead to see when/if I'm in

Cutting corners to meet arbitrary management deadlines



Essential

Copying and Pasting from Stack Overflow

O'REILLY®

*The Practical Developer
@ThePracticalDev*

The internet will make those bad words go away



Essential

Googling the Error Message

O RLY?

*The Practical Developer
@ThePracticalDev*

FINAL IS COMING!



- Material: ***Everything!***
- Homework, Labs, Lectures, Textbook
- **Wednesday, 3/21 in this classroom**
- **Starts at 8:00 AM **SHARP****
- ***BRING YOUR UCSB IDs PLEASE!***
Arrive 10-15 minutes early for seating changes
- Duration: **3 hours long** (but really designed for 1.5 – 2 hours)
- Closed book: no calculators, no phones, no computers
- Allowed: 1 sheet (**single**-sided) of written notes
 - Must be no bigger than 8.5" x 11"
 - **You have to turn it in with the exam**
- **You will write your answers on the exam sheet itself.**



Intro Stuff and For-Loops

Ch. 1 & 2

- What is CS? What are computers? Brief history
- What is programming? How does abstraction fit in?
- Representing Numbers and Using Arithmetic in Python
- Variables in Python
- Random Number Generation
- Loops using **for**
 - Differences between **for n in (...)** vs. **for n in range(...)**
 - Different uses of **range**
 - Implementing accumulations (**example: sum = sum + n**)

If-Else, Booleans, and Functions

- Conditional statements using **if/elif/else** Ch. 1 & 2
- Compound Boolean Logic
 - Example: What is `((a > c-d) or (b/c > a)) and (d > 1)`
- Functions – how to define them, how to call them
 - The difference between **print()** and **return**

Strings

Ch. 3

- Operations on strings:
Concatenation, Repetition, Indexing, `len ()`
- Member functions
(e.g. `string.center`, `.count`, `.lower`, `.index`, `.find`,
etc...)
- ASCII conventions (and functions `chr (n)` and `ord (c)`)

Lists

Ch. 4

- Lists and their member functions
(e.g.: `.append`, `.insert`, `.pop`, `.sort`, etc..)
- Lists operations
(e.g.: `max`, `min`, `len`, `sum`, creating lists of lists, etc..)
- Review the average, max/min, median algorithms

Dictionaries

Ch. 4

- Differences between dictionaries, tuples, and lists
- Member functions **.keys** and **.values**
- Operations on dictionaries
 - How do you create an **new** entry with a **key**?
 - How do you assign a **value** to a **key** entry?
- Review frequency counting examples we did using dictionaries
 - Modes and histograms example

File Input/Output

Ch. 5

- Why use file I/O?
- Opening and closing files
- Using for-loops to read a file
- Differences between `readline`, `readlines`, and `read`
- Reading HTML files over the Internet using `urllib.request`

Formatting Output Lines

- Using the `input()` function
 - What does that data type default to?
 - How do we force an input to be a non-default type?
- Using the `print()` function
 - How does the “,” operator work in there?
 - How does the “`end=`” option work?
- Converting one data type into another data type
 - Example: `x = str(66)` or `y = int("54")`
- Format modifiers using the “%” method
- Format modifiers using the `.format` method

While Loops, Control Structures, Digital Images

Ch. 5, 6

- Differences between **while** and **for** loops
- Ability to write the same loop in either fashion
- High-level control structures
 - Flow charts
 - What they tell us about how to best plan writing a program
 - No programming questions on this topic
- Differences between Raster vs. Vector graphics
- The RGB scheme and how it works in Python's `cImage` module using the `Pixel` class
 - *No programming* questions will be on this topic

Recursive Functions

Ch. 9

- How to write/interpret a recursive function
 - What are the 2 things you need to know to do recursion function programming?
 - If I give you a numerical sequence, make that into a recursive function.
 - Or if I show you a recursive function, tell me what it does

Homework, Labs, and Projects

- Review them ALL
and understand what you did

Sample Questions

What does this Python code print out?

```
n = 10
while (n > 4):
    print (n, end=".")
    n -= 1          # what is this?
```

10.9.8.7.6.5.

What does this Python code print out?

```
j = 1
while (j <= 5):
    print (j*5)
    j = j + 3      # can I write line this another way?
```

**5
20**

Re-write this code using only a for loop

Sample Questions

What does this Python code print out?

```
L = []
ct = 0
while (ct < 4):
    L.append(2*ct-ct/2)
    ct+=1
Print (L)
```

[0.0, 1.5, 3.0, 4.5]

What does this Python code print out?

```
k = 8
while (k < 10):
    print("While away!")
    for k in range(5, 13, 2):
        if (k == 7):
            print ("Lucky Seven!\n")
        else:
            print (k)
```

While away!

5

Lucky Seven!

9

11

Sample Questions

What does this Python program print out?

```
n = 1
m = 10
while (n < 12) or (m > 4):
    print(n + m, end=",")
    n += 5
    m -= 4
```

11,12,13,

How different would the answer be if we changed the “or” into “and”?

11,12,

Sample Questions

Write a Python function, **CollectNamesAges()**, that asks users to input names of people AND their ages that it will put in a dictionary *that it returns*. Users will be continually asked for names until they enter “END”. Ages must be stored as integer variables.

For example:

```
Please enter a name: Jim  
Please enter age for Jim: 30  
Please enter a name: END
```

When they do so, the function will *also print out* the dictionary. The string “END” must not be placed in the dictionary.

Answer to Previous Question

```
def CollectNamesAges():  
    D = {}  
    name = ""  
    while (name != "END"):  
        name = input("Please enter a name: ")  
        if name != "END":  
            age = int(  
                input("Please enter age for " + name + ": ") )  
            D[name] = age  
  
    print (D)  
    return D
```


Sample Questions

What does this Python program print out?

```
def Converter(dnary):
    newd = {}
    alist = (dnary.values())
    for item in alist:
        newd[item] = str((item-1)*2)
    return newd

Yums = {'crepe': 3, 'pho': 9, 'tabbouli': 10, 'roti': 9, 'guotie': 5}

print( Converter(Yums) )
```

{3: '4', 9: '16', 10: '18', 5: '8'}

Sample Questions

Write a **recursive** function in Python, **Sum(n)**, where **n** is a positive integer. The function returns the sum of the first **n** integers.

```
def Sum(n):  
    if n == 0:  
        return 0  
    else:          # else: in this example is optional  
        return n + Sum(n - 1)
```

</CS8>

**Best of Luck on
All of Your Finals**