Review for Final Exam

CS 8: Introduction to Computer Science, Spring 2019
Lecture #17

Ziad Matni, Ph.D.
Dept. of Computer Science, UCSB

Administrative

- Homework 8 due today!
- Lab 6 due today!

Final Exam Extra Review Session

Friday, June 7th

1:00 – 3:00 PM

PHELP 2510

(this is optional)

Finals Week

- Dr. Matni will have office hours on finals week

Monday 1:00 pm – 2:30 pm

FINAL IS COMING!

- Material: ***<u>Everything</u>*!**
- Homework, Labs, Lectures, Textbook
- **Tuesday, 6/11** in this classroom
- **Starts at 4:00 PM **SHARP****
- ***Bring your UCSB IDs and arrive 10-15 minutes early***
- Duration: **3 hours long** (but really designed for 1.5 – 2 hours)
- Closed book: no calculators, no phones, no computers
- Allowed: 1 sheet (double-sided ok) 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.**

STUDY GUIDE NOW ONLINE!

Example of Recursive Functions: **Linear Series**

- **Mathematical Linear Series**

Example:

S(n) = 0, 1, 4, 13, 40, . . .   for n = 0 to ∞

What's the pattern?

Linear series: $S_{n+1} = \mathbf{A}.S_n + \mathbf{B}$ where A & B are constants

In the example above: A = 3 and B = 1

What is our base-case?   **What is our recursion?**

Example: Linear Series

- **Mathematical Linear Series**

Example:

S(n) = 0, 1, 4, 13, 40, . . . for n = 0 to ∞

Linear series: $\underbrace{S_{n+1} = 3.S_n + 1}_{\textit{recursion}}$ and $\underbrace{S_0 = 0}_{\textit{base case}}$

```
def series(n):
    if n <= 0:
        return 0
    return (3*series(n-1) + 1)
```

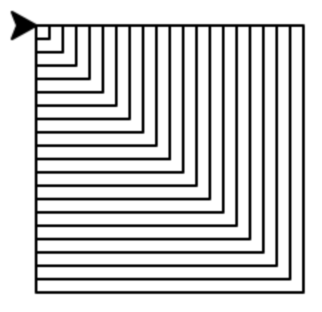
Example: Recursive Drawing

```
def drawS(aTurtle,side):
    for i in range(4):
        aTurtle.forward(side)
        aTurtle.right(90)

def nestedBox(aTurtle,side):
    if side >= 1:
        drawS(aTurtle, side)
        nestedBox(aTurtle, side - 5)
  # How come there’s no “base-case”??

import turtle
jim = turtle.Turtle()
nestedBox(jim, 100)
```

b/c side will eventually be < 1 and too small to draw



More Practice Questions! ☺

Exercise

What happens when we use the multiply operator on a list, like: **`[1, 2, 3] * 2`** ?

① I get [2, 4, 6]

[✓ ② I get [1, 2, 3, 1, 2, 3]]

③ I get [[1, 2, 3], [1, 2, 3]]

④ I get an error message

Exercise

How many times will the character “x” print in this code?

```
for m in range(3, 9, 2):
    print("x")
    n = m
    while (n < 7):
        print("x")
        n += 7
```

① 7 times

② 6 times

[✓ ③ 5 times]

④ 4 times

⑤ Infinite times

What is the exact output?

```
ucsb_classes = ['CS8', 'CS16', 'CS24', 'ECON1', 'COMM88',
'MATH3A', 'CHEM6A']
l = []
# Note that: chr(65) = ‘A’
for c in ucsb_classes:
    if c[0] == chr(67):
        l.append(c.lower() + "!")
print(l)
```

Exercise

Write a Python function, **AddG(s)** that takes a string **s** as a parameter and returns a string with “g” after each character in the original string.

For instance:
if **s="abcd"** then, **AddG(s)** becomes **"agbgcgdg”**, or
if **t="aog"** then, **AddG(t)** becomes **"agoggg”**,

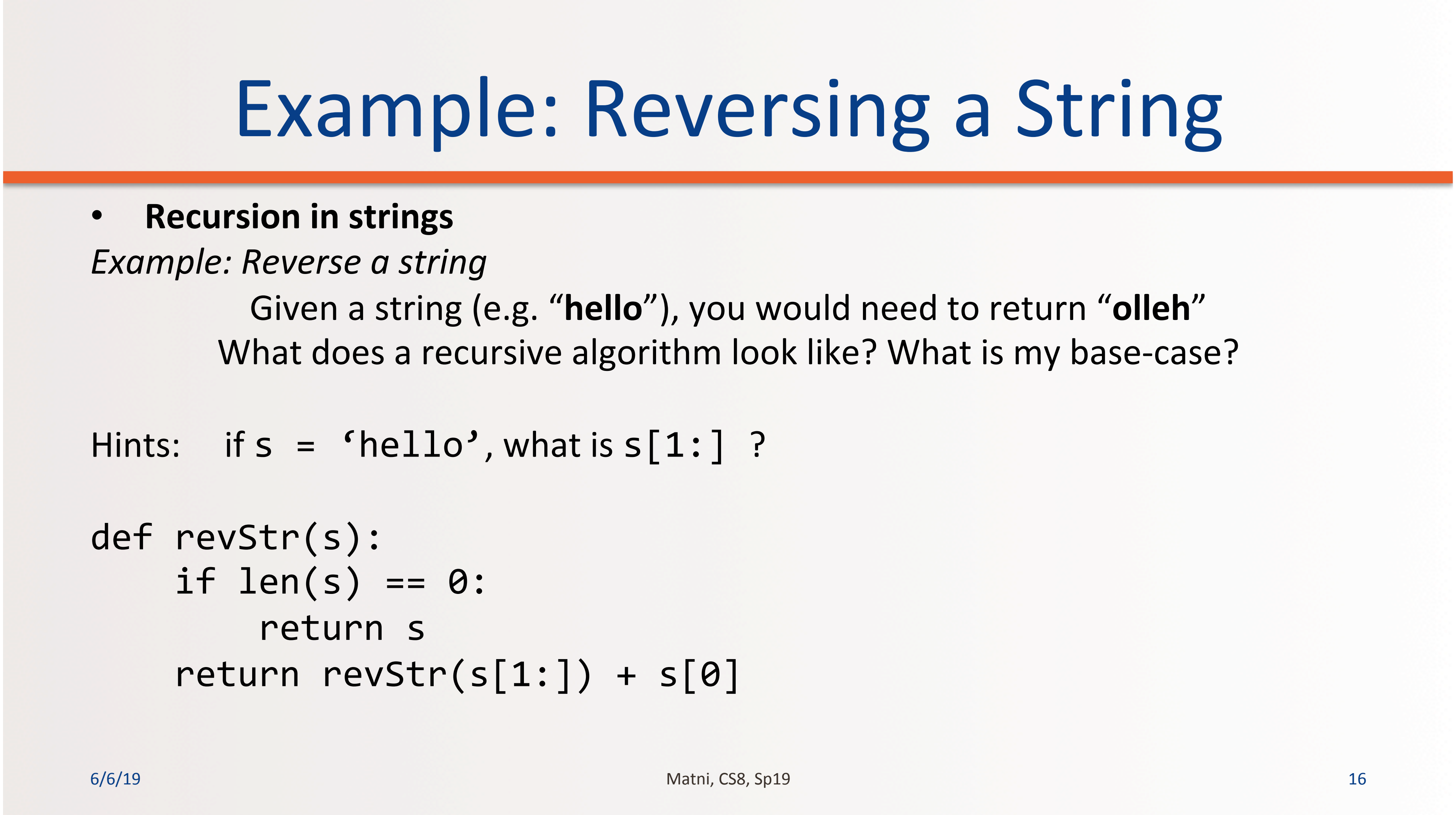
Exercise

- **Given a dictionary:** `D = {1:0.1, 2:0.2, 3:0.3, 4:0.4}`
- **What does this code do?**

```
D2 = {}
L = list(D.values())
k = 5
for p in L:
   D2[k] = 1+p
   k += 2
print(D2)
```

```
It prints:
{5: 1.1, 7: 1.2, 9: 1.3, 11: 1.4}
```

</LECTURE>



Example: Reversing a String

- **Recursion in strings**

Example: Reverse a string

Given a string (e.g. “**hello**”), you would need to return “**olleh**”
What does a recursive algorithm look like? What is my base-case?

Hints: if `s = ‘hello’`, what is `s[1:]` ?

```
def revStr(s):
    if len(s) == 0:
        return s
    return revStr(s[1:]) + s[0]
```