On your way in...

Pick-up
1. HW5, due Wednesday
Welcome to CS 134!

Introduction to Computer Science
Iris Howley

-Recursion, Classes-

Spring 2019
James Mickens, Class of 60s’ Scholars Lecture

QUESTION & ANSWER SESSION

Friday/TODAY, 3/8 at 2:35pm in Wege

Submit your questions!
A Thought.
WE ARE LEARNING MANY THINGS IN THIS CLASS

1. python
2. How to think like a computer scientist
   • See the textbook title: “Think Python: How to Think Like a Computer Scientist”
Some Examples

for year in range(2010,2019+1):
    print(“The year is”, year)

for (year in 2010:2019) {
    print(paste("The year is", year))
}

for (int year=2010; year<=2019; year++) {
    System.out.println("The year is "+ year);
}
RECURSION IS A FUNCTION CALLING ITSELF. IT MAKES SOME SUCCINCT SOLUTIONS AND INTERESTING DATA STRUCTURES.

(Today’s lesson)
Dragon, I need to know if any of the numbers in this list are odd:

(3142, 5798, 6550, 8914)
Sorry, I can only tell you if the *first* number of the list is odd.
But I need to know if *any* number in the list is odd, not just the first!
I’ll only look at the first number, but I’ll look at as many lists as you like.
What should Sam do?
The first number is not odd.
(3142, 5798, 6550, 8914)

The first number is not odd.
(3142, 5798, 6550, 8914)

The first number is not odd.
(3142, 5798, 6550, 8914)

The first number is not odd.
That’s an empty list! It can’t be odd.
None of the numbers the Sorcerer gave me were odd, thank you!
How can you know that? I only told you about the first number!
The lists I gave you were:
(3142, 5798, 6550, 8914)
(5798, 6550, 8914)
(6550, 8914)
(8914)
()

Tricky. Looks like you’ve discovered recursion.
The lists I gave you were:
(3142, 5798, 6550, 8914)
(5798, 6550, 8914)
(6550, 8914)
(8914)
()

Why did this work?
The lists I gave you were:
(3142, 5798, 6550, 8914)
(5798, 6550, 8914)
(6550, 8914)
(8914)
()
Steps for Recursion

• Know when to stop.
• Decide how to take one step.
• Break the journey down into that step plus a smaller journey.
Steps for Recursion

• When to stop?
  • When list is empty

• What is the one step?
  • Check the first list item

• How to break the journey down?
  • Progress through each of first list items
Pseudocode

Sam: The list is \{3142, 5798, 6550, 8914\}
Dragon: Is list empty?
Dragon: No? Is first number odd?
Dragon: Yes? Print.
Sam: Drop first num from list!
Dragon: Is list empty?
Dragon: No? Is first number odd?
Dragon: Yes? Print.
Pseudocode

mylist = [3142, 5798, 6550, 8914]
printFirstOdd(mylist)

def printFirstOdd(l):
    is l not empty?
    is l[0] odd?
    print l[0]
    check the rest of our list
Python

mylist = [3142, 5798, 6550, 8914]
printFirstOdd(mylist)

def printFirstOdd(l):
    if l:
        if l[0] % 2 != 0:
            print(l[0])
        printFirstOdd(l[1:])
def zappa(s):
    """Computes a function of string s."""

    if not s:
        return 0
    else:
        return zappa(s[1:]) + 1
Fibonacci Numbers

(we’ve seen this before)

def fibo(n):
    if n < 2:
        answer = n
    else:
        answer = fibo(n-1) + fibo(n-2)
    return answer
Iteratively Computing Factorial

• 5!
  - \(5\times4\times3\times2\times1 = 120\)

• `def factorialIt(n):
  \[\text{result} = 1\]
  \[\text{for } i \text{ in range}(2, n+1):
  \quad \text{\$result * } i\]
  \[\text{return result}\]
Recursively Computing Factorial

• 5!
  ▪ 5*4*3*2*1 = 120

• def factorial(n):
  ▪ if n <= 1:
    o return 1
  ▪ else:
    o return n * factorial(n-1)
CLASSES ENCAPSULATE INFORMATION SO WE DON’T HAVE TO KEEP TRACK OF EVERYTHING

(Today’s lesson)
```
__slots__ = []

>>> class Yesteryears:
...     """demo of classes from last week """
...     ...
>>> yy = Yesteryears()
>>> yy.start = 2018
>>> yy.end = 2022
>>> yy.mid = 2020
>>> yy.whatev = "I do what I want!"

>>> class Years:
...     __slots__ = ['start', 'end']
...     ...
>>> newy = Years()
>>> newy.start = 2017
>>> newy.end = 2021
>>> newy.mid = 2019
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'Years' object has no attribute 'mid'
```
Classes

```python
>>> from oracle import Oracle
>>> o = Oracle()
>>> type(o)
<class 'oracle.Oracle'>
```

- `o` is an instance of the class, `Oracle`
- `Classes` are user-defined types
Classes

>>> Class Years:
    """ Define some attributes """
>>> y = Years()
>>> y
    <__main__.Years object at 0x108c63860>
>>> from oracle import Oracle
>>> o = Oracle()
    This is __name__!!!!
>>> o
    <oracle.Oracle object at 0x103485e48>
Classes

See the Textbook, Chapter 15.

Point and Rectangle classes!
QUESTIONS?
Leftover Slides
Fibonacci Sequence

- \(\text{fibo}(0) = 0\)
- \(\text{fibo}(1) = 1\)
- \(\text{fibo}(n) = \text{fibo}(n-1) + \text{fibo}(n-2)\)

- \(\text{fibo}(4) = \text{fibo}(3) + \text{fibo}(2)\)
  - \(= \text{fibo}(2) + \text{fibo}(1) + \text{fibo}(1) + \text{fibo}(0)\)
  - \(= \text{fibo}(1) + \text{fibo}(0) + 1 + 1 + 0\)
  - \(= 1 + 0\)
  - \(= 3\)
Speeding Up Fibonacci

(Memoization)

global postit
if n in postit:
    answer = postit[n]
else:
    if n < 2:
        answer = n
    else:
        answer = fibo(n-1) + fibo(n-2)
    postit[n] = answer
return answer