On your way in...

Pick-up:
1. Graded Homework 5
2. Lecture #20 Notes
Welcome to CS 134!

Introduction to Computer Science
Iris Howley

-Linked Lists-

Spring 2019
Do you want to be a Computer Science Teaching Assistant this fall?

• TA applications due April 12 (next Friday)

• [https://csci.williams.edu/tatutor-application/](https://csci.williams.edu/tatutor-application/)

• You’ll need 2 faculty references
  • Duane or Iris should be at least one of those!
  • Come talk to us!
CS134 Final Exam

• Our final exam is scheduled for Sunday, May 19 at 1:30 PM

• If you have a conflict, please send me an email by Monday, April 8 explaining what that conflict is

• We have a limited number of slots available for an alternative session
  • Which will likely be on Tuesday May 14, during Reading Period
MAKING OUR OWN DATA STRUCTURES

Building our own classes, leveraging recursion, etc.
What is a list?

```python
class Element:
    _value
    _next
```

```plaintext
_3
_next

_15
_next

_2019
_next
```
class LinkedList:

    _head

class Element:

    _value

    _next

    _value

    _next

    _value

    _next
Linked List

• Class LinkedList is a “wrapper class” for our “container class”, Element

• This implementation, LinkedList mostly:
  1. Handles the empty case
  2. Passes the heavy-lifting (i.e., all other cases) to Element
QUESTIONS

(you’re not the only one!)
Linked Lists

- See example code in shared/examples/04/05!

- Lecture notes from 4/3 and 4/5 are also useful!
From Interactive Python

```python
>>> from LinkedList import *
>>> ll = LinkedList()
>>> ll.append(11)
>>> ll[0]
11
>>> ll[0] = 92
>>> ll[0]
92
>>> mylist = [0,1,2]
>>> mylist[2]
2
>>> e = Element(42)
>>> e.next = Element(99, Element(1000))
>>> str(e)
'Element(42), next=Element(99, Element(1000))'
>>> e.value
42
>>> ll.append(e)
>>> ll[0].value
99
>>> ll[0].next.next.value
1000
>>> bool(None)
False
>>> e
<Element object at 0x101d8e1d0>
>>> str(e)
'Element(42), next=Element(99, Element(1000))'
>>> e.value
42
>>> e.next.value
99
>>> e.next.next.value
1000
>>> bool(e)
True
```

```python
>>> ll = LinkedList()
>>> ll.append(5)
>>> str(ll)
'[5]'
>>> ll.append(700)
>>> str(ll)
'[5,700]'
>>> ll.append("hello")
>>> str(ll)
'[5,700,hello]'
>>> ll = []
>>> ll.extend([5,4,3])
>>> 3 in ll
True
```
QUESTIONS?
Leftover Slides
Tuples, Strings, other built-in types aren’t particularly special!

You can build your own!
What is a list?

class `list`(object)

    list() -> new empty list
    list(iterable) -> new list initialized from iterable's items

Methods defined here:

    `__add__`(self, value, /*)
        Return self+value.

    `__contains__`(self, key, /*)
        Return key in self.

    `__delitem__`(self, key, /*)
        Delete self[key].

    `__eq__`(self, value, /*)
What is a list?

3 → 15 → 2019
What is a list?

What is the last elephant holding onto?

None
Thought question:
How would you build a doubly-linked list?