Objectives

- Computer Science is Complexity Science
- Course logistics
- BI: Facebook
- Review for Final

1

Review

- What is recursion?
  - ➢ Provide an example of solving a problem recursively
- What are characteristics of programming languages?
- What are common constructs in programming languages?
- What are some differences between programming languages?

2

1

Review: Recursive Binary Search

```
def search(searchlist, key):
    if len(searchlist) == 0:
        return -1
    mid = len(searchlist)//2
    if searchlist[mid] == key:
        return mid
    elif key > searchlist[mid]:
        # look in upper half
        return search( searchlist[mid+1:], key )
    else:
        # look in lower half
        return search( searchlist[:mid], key )
```

Base case: We know the key is not in our list

**Recursion**

3

Review: Recursion Summary

- *Recursion*: method of solving problems
  - Break a problem down into smaller subproblems until problem is small enough that it can be solved trivially
- Binary Search:
  - Break problem to ~half the size of original problem
  - Base cases: when the middle element is what you're looking for; when there are no elements in your list
- Any recursive problem can be solved iteratively
  - Some problems lend themselves better to recursive solutions

4

2
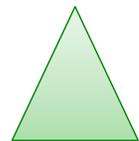
Review: Programming Language Characteristics

- **Syntax**: symbols used

- **Semantics**: what the symbols *mean*

5

Review: What is Computer Science?

"Computer Science is no more about computers than astronomy is about telescopes."
                                    --Edsger Dijkstra

6

3

> A human must turn information into intelligence or knowledge. We've tended to forget that
>   **no computer will ever ask a new question.**
>   -- Grace Hopper

> Computers are incredibly fast, accurate, and stupid.
> Human beings are incredibly slow, inaccurate, and brilliant.
> Together they are powerful beyond imagination.
>   -- Albert Einstein

7

Review: What This Course Is About

Problem Solving!

From
30 Rock

8

4

Review: Parts of an Algorithm

- Input, Output
- Primitive operations
  - ➤ What data you have, what you can do to the data
- Naming
  - ➤ Identify things we're using
- Sequence of operations
- Conditionals
  - ➤ Handle special cases
- Repetition/Loops
- Subroutines
  - ➤ Call, reuse similar techniques

An overview for the semester!

9

COMPLEXITY SCIENCE

10

CS == Complexity Science

- How can it be done?
  - Based on **information**
  - Managing, manipulating data
  - Possible algorithms
- How well can it be done?
  - Most **efficient** algorithm in terms of time and/or space
- Can it be done at all?
  - Often, proof is a program--an implementation of the above

11

Computer Science != Programming
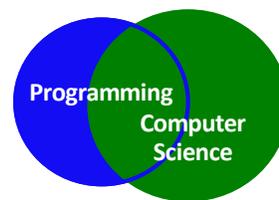
programming : CS ::

machining : engineering

grammar : literature

equations : mathematics

walking : W&L

a vehicle, not a destination

12

Computer Science Fields

Systems
- Architecture
- Operating systems
- Networks
- Distributed and parallel systems
- Databases
- Security
- …

Software
- Compilers
- Graphics
- Software engineering
- Software testing and verification
- …

Theory
- Algorithms
- Theory of computation
- …

Other
- Artificial intelligence
- Robotics
- Natural language processing
- Bioinformatics
- Visualization
- Numerical analysis
- …

- Often research involves combinations of these fields
- Not just programming!
  - ➢ But programming is a tool to do much, much more!

13

Computer Science Fields

Systems
- Architecture *
- Operating systems *
- Networks *
- Distributed * and parallel systems
- Databases
- Security
- …

Software
- Compilers
- Graphics *
- Software engineering*
- Software testing* and verification
- …

Theory
- Algorithms *
- Theory of computation
- …

Other
- Artificial intelligence *
- Robotics *
- Natural language processing *
- Bioinformatics
- Visualization*
- Numerical analysis
- …

* = field we discussed or did a problem in
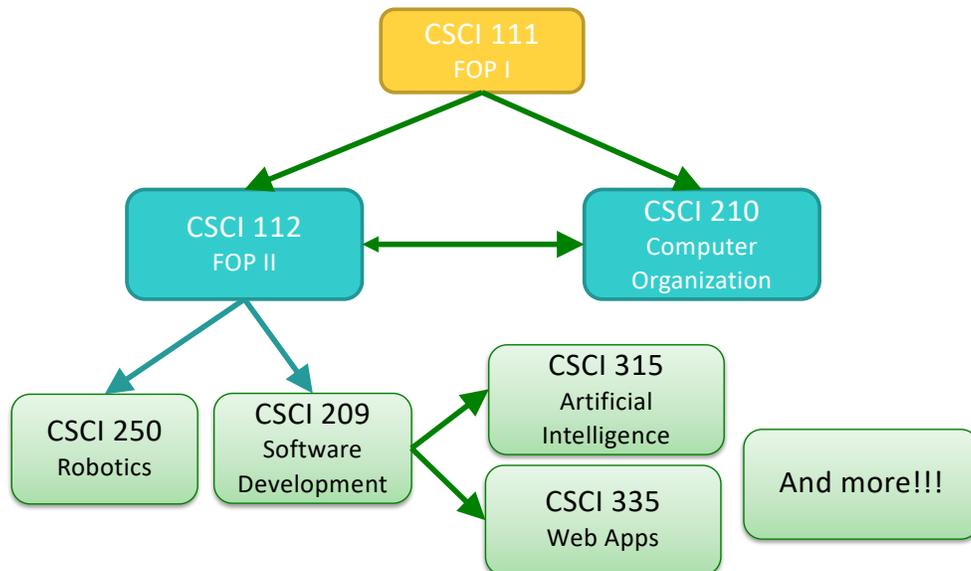 - ➢ Some are a stretch :)

14

Where Can You Go from Here?

15

Course Conclusions

- Better [computational] problem solver
- See impact of computer science on your life
  - Think differently about issues
- Understand some computing issues better
  - Taking out some of the mystery
  - Testing, debugging, efficiency
- Algorithms are everywhere
  - Process for solving problems, **efficiently**
  - Mapping human intuition to systematic/automatic process

16

Final: Broader Issues

- Take-home (untimed, open articles) part on Broader Issues
  - ➢ Submit PDF on Canvas by Finals Friday at noon
- 20 points of the final exam
  - ➢ Exam is worth 220 points

17

Final: The Rest

- Timed part on Canvas
- Only open brain, Canvas, Word
- Closed everything else
  - ➢ Turn off notifications, hide distractions
- Can have paper for scratchwork
- Some questions "in" Canvas
- Some questions in a Word document

18

Final: Word Part

- One question in Canvas has the Word document
- Download document, type your answers in document
  - I only left a few lines between questions
  - Write your answer below/between the questions
  - Use the point amount to help gauge how much to write
  - Be careful about autocorrect (e.g., avoid i as a variable)
- Submit Word document

19

Final Exam Review

- Focus on object-oriented programming
- New content: search techniques, lists (1D and 2D), complexity science
- Cumulative:
  - Functions, data types, common methods & operations
  - How to model data

Your questions?

20

Course Evaluations

- On Canvas, due Monday

- Incentive

  - If 60% of students complete evaluation,
    1% Extra Credit on lab grades

  - For each additional 10% of students who complete
    evaluation, 1% additional EC on *lab* grades

  - Total possible EC: 5%

21

Extra Credit Opportunity

Evaluate up to 2 games for up to
10 points Extra Credit towards labs
- On Canvas
- Submit image of feedback card
  or complete questionaire

22

Broader Issue: Facebook

Declan Jenna Lakpa Mary Patrick	Amanda Cassandra Han Mac Renan	Ellie Ford Jack Shelby	Aiden Cole Matt Nick

23

Broader Issue: Facebook News Feed

- How does Facebook's News Feed work?
  - How is the problem broken down?
  - What is its scale?
  - Is it what you expected?
- React to the statement, "In a large complex system like this, bugs are inevitable and understandable"
- What is the future of social network sites and their algorithms?

24

Final Exam Review

- What is our process for developing classes?

- What are the different ways to iterate through a list?

- How do you iterate through a dictionary?

25

Animal Shelter Software

- We want to keep track of animals at an animal shelter

What is our process for developing a class?

26

Process

- Determine data, functionality
- Create class
  - Create __init__, __str__ methods
- Test
- Create additional methods, test

27

Class: Pet

- Data:
  - Species of animal (dog, cat, chinchilla)
  - Name
    - Defaults to ""
  - Status (in holding, in adoption room, adopted)
    - Defaults to "in holding"
- Functionality
  - Constructor: `Pet(species)`
  - String format: `"species: name, status"`
  - Setters for name
  - Set animal as adopted or in adoption room
  - Getters for this information

28

Counter Class Specification

- A class that represents a counter that wraps around from a high value back to its low value
- Functionality:
  - Constructor – takes as parameters the low value and the high value
    - counter starts at low value
  - A string representation of the Counter
    - e.g., "low: <low> high: <high> current: <current>"
  - Increment the counter by a given amount (a positive amount), wrapping around to low again, if necessary. Returns number of times had to wrap around.
  - Decrement the counter by a given amount (a positive number), wrapping around to high again, if necessary. Returns number of times had to wrap around.
  - Sets the counter's value, only if low <= value <= high. Otherwise, prints an error message.
  - Getters: low, high, current value

29

Palindrome

- Write a program that determines if a string (input by a user) is a palindrome. A *palindrome* is a word that is the same forwards and backwards. Some example palindromes: "kayak", "A man A plan A canal Panama".
- http://www.fun-with-words.com/palin_example.html
- Break the problem into at least two functions:
  - main
  - isPalindrome, which returns True iff the parameter string passed into the function is a palindrome.
- Depending on how you think about the problem, you may want to break the solution into more functions, e.g., a reverseString function

30

Make Good Decisions!

31