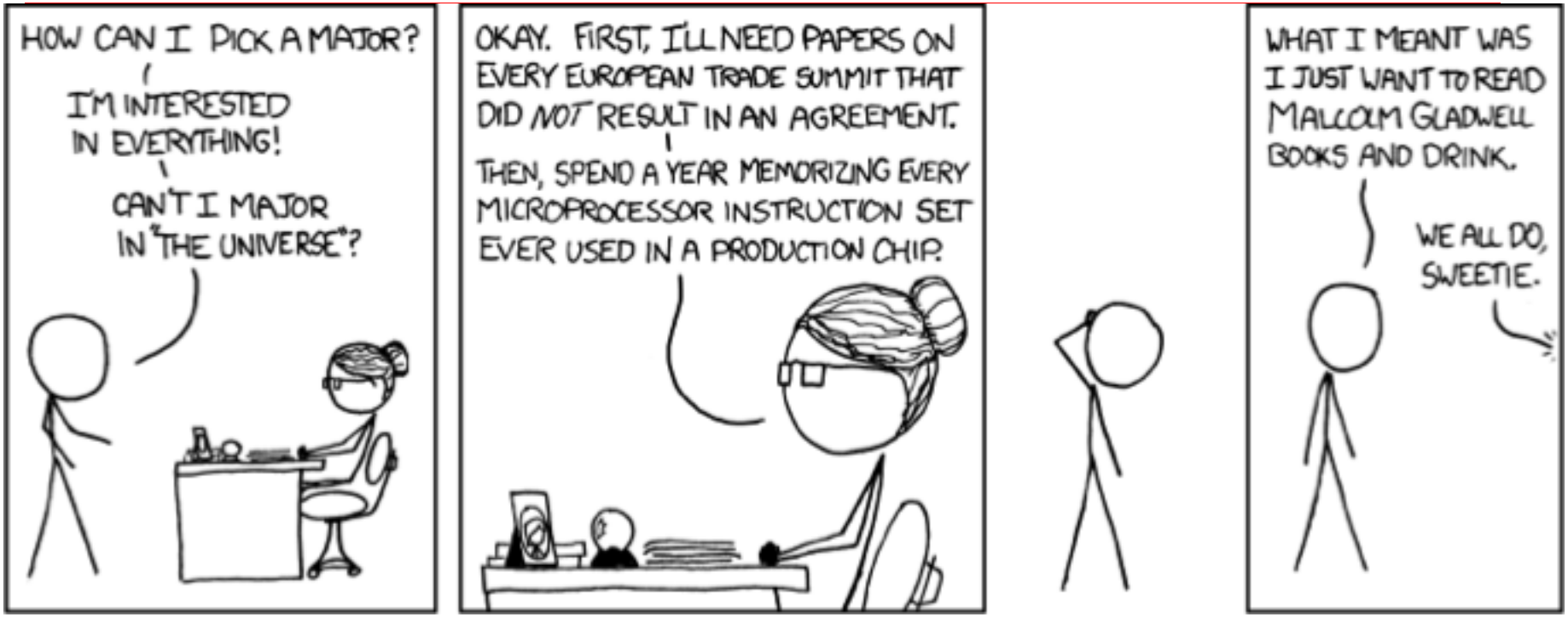
CS 111 (Law): Program Design I
Lecture 1: Introduction

Robert H. Sloan & Richard Warner
University of Illinois at Chicago
Tuesday, August 27, 2019



Randall Munroe, XKCD, http://xkcd.com/863/

FIRST: WHICH COMPUTER SCIENCE CLASS?

UIC CS Classes

- CS 100, Discovering Computer Science:
  - CS for fun, no credit if in College of Engineering
  - MWF 9am; LC F3; has open seats
- CS 111, Program Design I (This course!)
  - Students with ***little to no*** programming experience who
    - Are CS majors or minors, or CoE students planning to use CS 111–141, or any students wanting to learn more
    - Now in 2–3 flavors! (This is the “law” flavor)
- CS 141, Program Design II: If you already have programming experience

2 Versions of CS 111 Fall 2019

- ***Law Themed:* 2:00–3:15 pm, LC B1**
  - Team taught by Profs. Robert Sloan (UIC CS) and Richard Warner (Chicago-Kent Law School)
  - Probably slightly more total work
- *Media Themed (UIC Traditional):* LC B1, MWF 12:00–12:50.
  - CS Prof. Hayes

What will everybody do in CS 111?

- Learn the basics of programming in Python
  - Very popular programming language, commonly used
  - Easy to read and understand
  - Great for CS1 because can see realistic examples earlier
  - More about “Why Python” soon
- And…

One of (based on section)

- Use media computation
  - Draw pictures, manipulate images, manipulate sound files
- Investigate legal and privacy issues
  - E.g., build heat maps of Chicago gun crime hot spots and discuss Chicago's predictive policing; write encryption & discuss crypto law/policy

Comparing 111 sections

- Reasons to be in this section: Opportunity
  - Integration of traditional CS 111 material with timely topics on law, policy and society, such as crime rates and cryptography.
  - Team taught class, with ~20% of class material presented by law school professor.
  - Learn some data science in addition to software development.
- *But*: Probably mildly more work; hassle a little more with dirty real-world data

What if I already have programming experience? **CS 111 Proficiency Exam**

- (Should already have heard about this at orientation!)
- Steps: (Note must be CS major)
  - Log into my.uic.edu portal
  - Click UIC Connect/Students
  - Click Required Placement Tests
  - Select CS 111 Proficiency exam
- Demonstrate ability to write short program
  - Language should be “C-derivative” (Python, C, C++, Java, PHP, etc.)
- Should show you understand: Variables, loops, if statements, arrays (or perhaps lists if Python), function calls

Mrs. Mutner liked to go over a few of her rules on the first day of school.

NEXT:
A LITTLE BORING LOGISTICS STUFF

Instructors: Robert Sloan & Richard Warner

- Sloan: UIC CS Prof.
  - Current Research: Policy & Legal issues relating to electronic security & Privacy and AI fairness; Computer Science Education
  - In past: Theoretical CS, AI, Cryptography, Security

- Warner: IIT Chicago-Kent Law School Prof.
  - Expertise: Contracts, Philosophy, *E-Commerce, Internet, and Privacy Law*

Office Hours: Please come!

Sloan

- SEO 1112
- Tue 3:30–5:00
  - This week only: Thu 11:00–12:30
  - Will almost certainly need to move day couple times later on
- And by appointment

Warner

- 845 Chicago-Kent Law School, Adams St. (about 1.2 miles)
- Tuesday, 3:45–4:15
- And by appointment

TA's. (Office hours; *locations* TBA)

Krisnha Garg
kgarg8@uic.edu
Tue and Thur 12–1 pm

Xu Lin
xlin48@uic.edu
Mon, 12–4

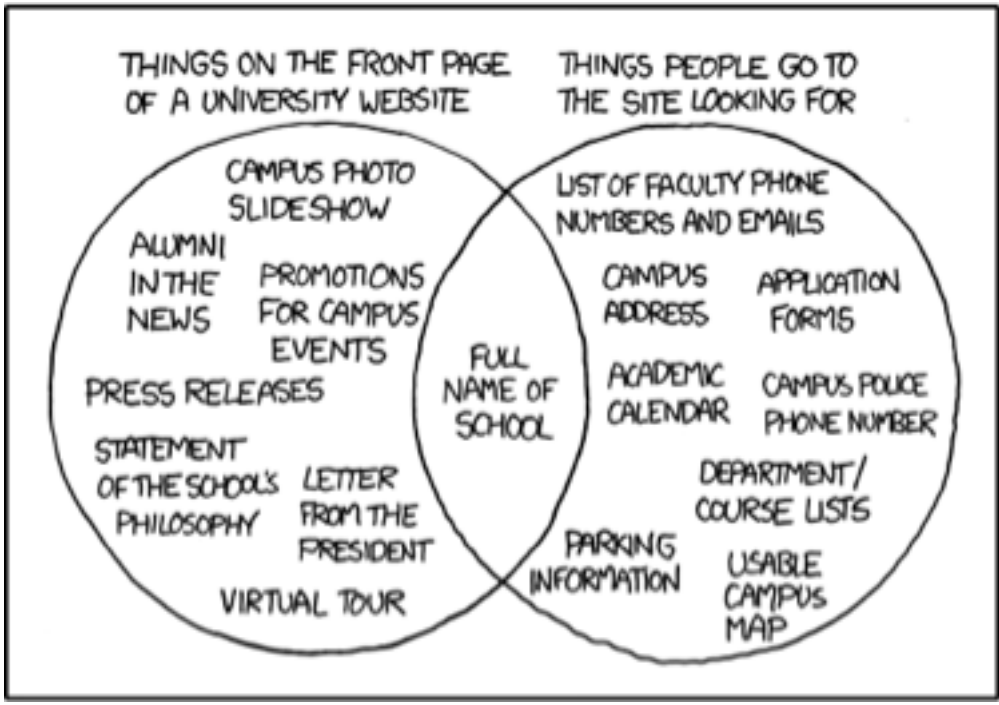
Zixuan Ke
zke4@uic.edu
Thu 9–11, Fri 3–5

Mobashir Sadat
msadat3@uic.edu
Tue 11–2 & Thur 11–12

Piazza!

Piazza!

Everything is on some website



- Zybooks:
  reading, **assignments**
  https://learn.zybooks.com/
- Blackboard: grades
  blackboard.uic.edu
- Course website:
  Syllabus, schedule, list of readings, lecture slides
  cs111law.class.uic.edu
- Piazza Forum (more next slide)

Use Piazza instead of emailing

piazza.com/uic/fall2019/cs111law

- For public questions (lab and project questions)
  - Your classmates benefit from your questions
  - Your classmates can answer your questions
  - We will check the forum frequently

- For personal questions (grades)
  - Can ask questions to instructors only
  - Means any of us (Profs & TAs) can answer your question
  - All CS 111 communication is in one place

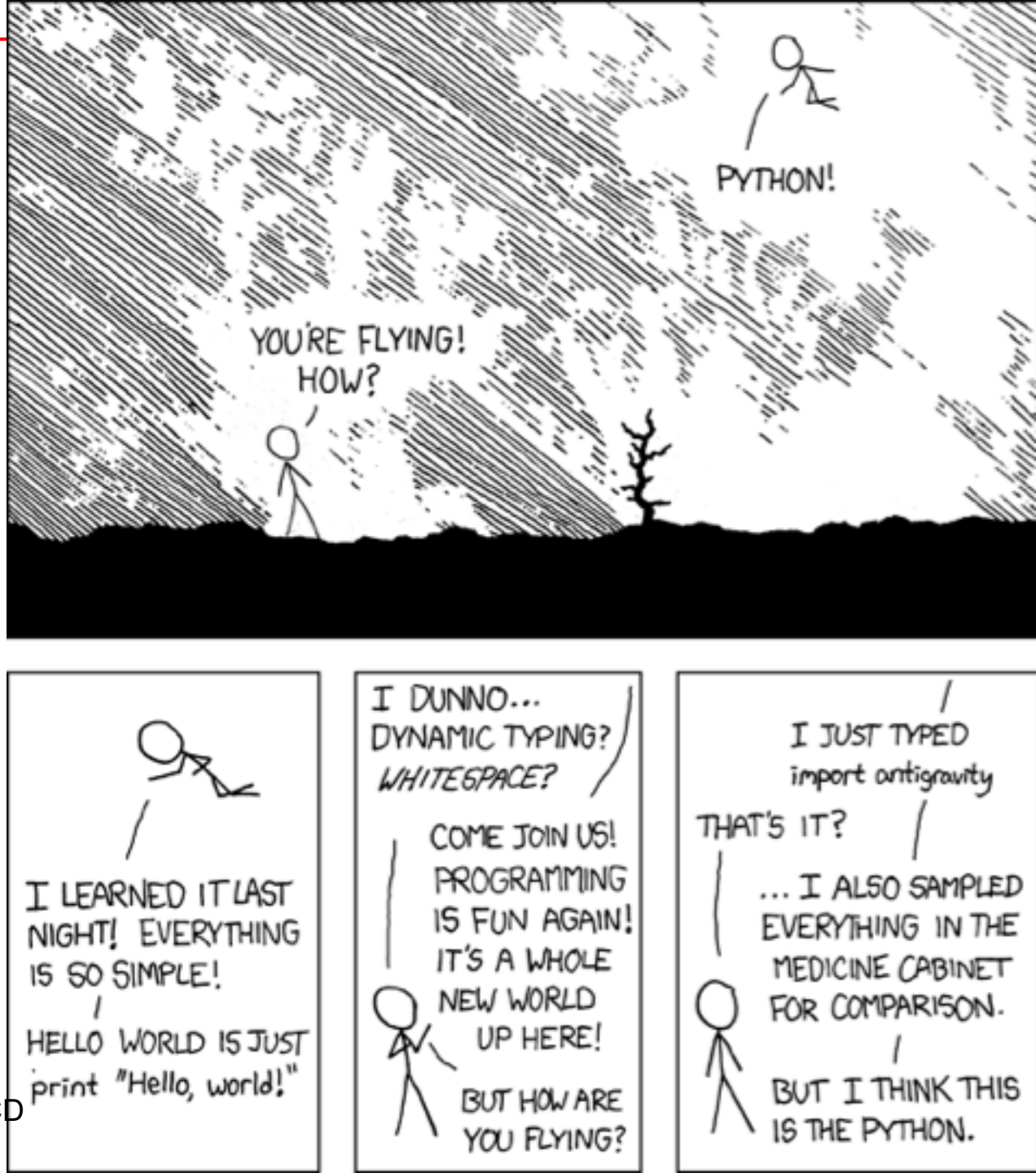
Clickers!

- Lets you vote on multiple choice questions in real time.
- Like pub trivia, except the subject is always CS 111
- **Buy at bookstore unless you already have; register on blackboard before next class!**

Register your clicker online so we can give you points

- In Blackboard, on the side of the course page
- Your votes will be saved before you register, just not associated with you

This class uses Python



Randall Munroe, XKCD
http://xkcd.com/353/

A fast taste of Python

- In Spyder environment which we'll use for most of class
- (Tomorrow's lab all inside the Zybook)

Course "Textbook"

- *Programming in Python3 with Zylabs,* https://learn.zybooks.com/ Sign-up code **UICCS111Fall2019,** price: $58.
- *How to Think Like a Computer Scientist: Interactive Edition,* a few portions only: http://interactivepython.org/runestone/static/thinkcspy/index.html
  - Covers introduction to Python programming
- Some assigned readings on legal topics

Before Next Class (by 1:30pm Th)

- Register for Piazza (if we didn't already register you)
- *Register your clicker on blackboard*
- *Read Zybooks Intro to Python 3: 1.1–1.4, 1.6–1.0*
  - *Do all participation activities*.
- Read How to Think Like a Computer Scientist: Interactive Edition, from General Introduction, 1.1–1.2, 1.5, 1.11.
- If you want to get ahead: Install Python **using our instructions (course website, resources)** on your machine (else in lab tomorrow or next Wed.)

Randall Munroe, XKCD, http://xkcd.com/1425/

INTRODUCTION TO COMPUTER SCIENCE (CS)

What is computer science?

- Biology = study of living things
- Chemistry = study of structure and composition of matter
- What is computer science the study of?

What is computer science?

- Biology = study of living things
- Chemistry = study of structure and composition of matter
- What is computer science the study of?
- Computer Science $\neq$ study of computers!

What is computer science?

- Biology = study of living things
- Chemistry = study of structure and composition of matter
- What is computer science the study of?
- Computer Science $\neq$ study of computers!
- Computer Science is primarily study of:
  - Processes $\approx$ Algorithms $\approx$ Programs $\approx$ Recipes

Some Pioneers' 1967 Vision of CS

"**The theory and design of computers,**

as well as

the study of all the phenomena arising from them"

Allen Newell, Alan Perlis, Herb Simon, *Science,* 1967

Some Pioneers' 1967 Vision of CS

"**The theory and design of computers,**

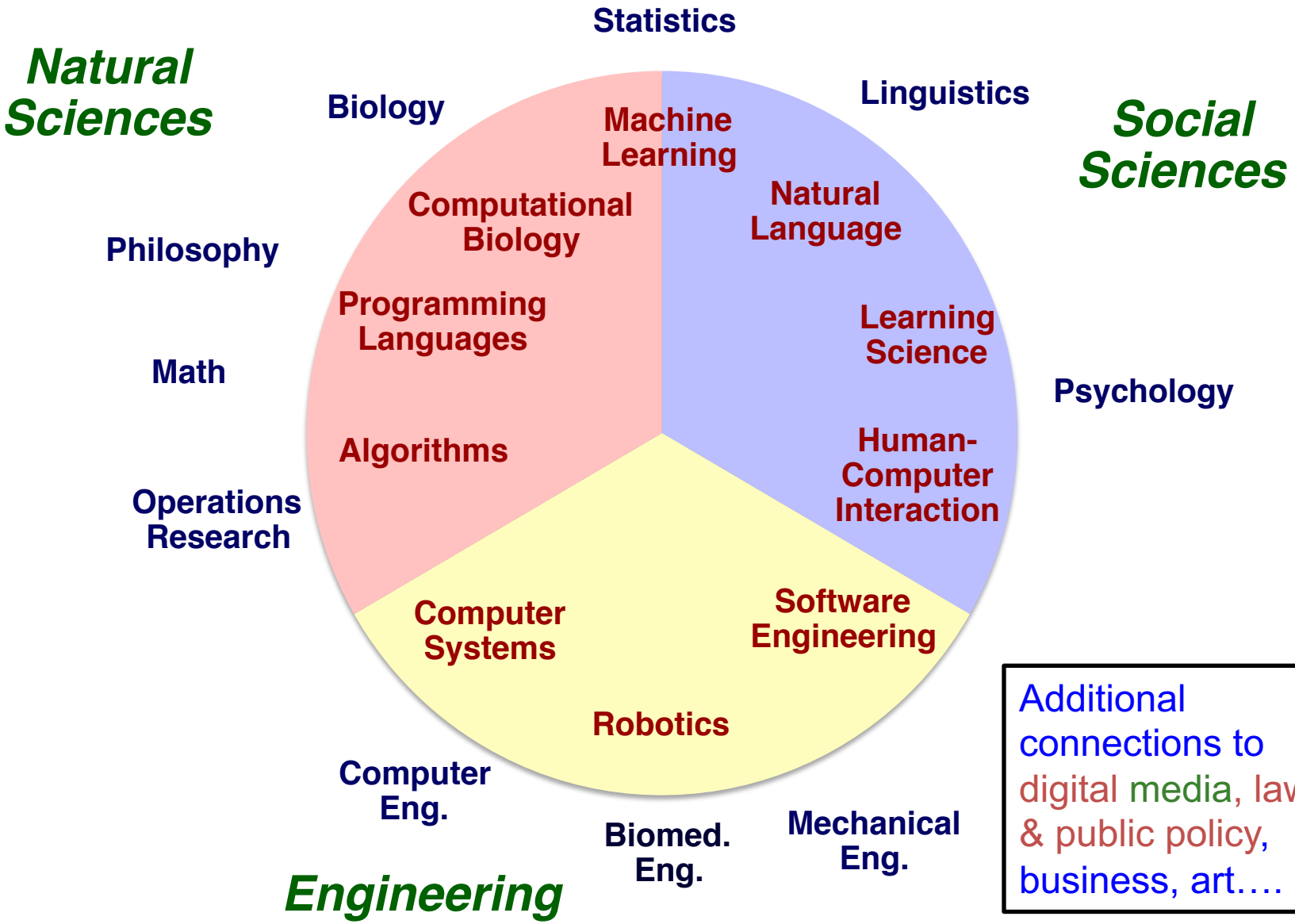
as well as

the study of all the phenomena arising from them"

Allen Newell, Alan Perlis, Herb Simon, *Science,* 1967

Only really big difference today: Design of computers is its own field: *Computer Engineering*

CS's Disciplinary Roots today



Additional connections to digital media, law & public policy, business, art….

What about programming?

- Computer science ≠ programming!
- But lawyers, researchers, business executives, computer scientists, etc., all study writing to communicate for at least 13 years
- Programming is other main way computer scientists communicate; requires extensive study

What about programming?

- Computer science ≠ programming!
- But lawyers, researchers, business executives, computer scientists, etc., all study writing to communicate for at least 13 years
- Programming is other main way computer scientists communicate; requires extensive study
- Almost all computer scientists spend some of their working time programming just as almost all lawyers spend some time writing

What about programming?

- Computer science ≠ programming!
- But lawyers, researchers, business executives, computer scientists, etc., all study writing to communicate for at least 13 years
- Programming is other main way computer scientists communicate; requires extensive study
- Almost all computer scientists spend some of their working time programming just as almost all lawyers spend some time writing
  - One ironic exception: Some CS professors!

If Computer Science

- is the study of computation, meaning
- Processes ≈ Algorithms ≈ Programs ≈ Recipes
- Then the next question is:

What's computation good for

- Computer science is the study of recipes
- Computer scientists study…
  - How the recipes are written (algorithms, software engineering)
  - The units used in the recipes (data structures, databases)
  - What can recipes be written for (systems, intelligent systems, theory)
  - How well the recipes work (human-computer interfaces)

Key concept:
The *COMPUTER* does the recipe!

Make it as hard, tedious, complex as you want!

- Look through 100,000 documents in a lawsuit to pick out the 17 or the 42,000 containing a particular phrase? Easy-peasy!
- Crank through a million genomes? No problem!
- Find one person on UIC's 30,000 student campus? Yawn!
- Process a million dots on a screen or a bazillion sound samples?
  - (What the media computation course will do)

Heading to Python Programs?

2 eggs
1 c milk
1 c sifted flour
½ t vanilla extract
12 oz. farmer's cheese
4 oz. cream cheese, softened
⅓ c sugar
1 t lemon juice
1 t vanilla extract
1 egg yolk

Batter: Combine eggs, milk, salt and vanilla and blend well. Gradually add flour. Beat well.

Filling: Combine cheeses, sugar, juice, vanilla and egg yolk in a bowl and mix together until smooth.

Repeat until all the batter is used:

Put a ladleful of batter into heated greased skillet. Tilt pan to swirl the batter so it covers the bottom of the skillet. Fry on one side until bubbles form, the top is set, and bottom is golden brown. Carefully loosen edges of the crepe and slip it out of the skillet onto a plate.

Fill each fried crepe:

Place 1 tablespoon of filling on one edge. Roll once to cover filling. Fold sides into the center and continue rolling until completely closed.

After all are assembled, heat 2 tablespoons of butter in the skillet and place each crepe, seam side down, in the skillet and fry 2 minutes on each side, turning once.

Features of recipe

- Fixed vocab of words, abbreviations, symbols
- Syntax: Rules about what can be said & where
- Sequence of operations to be done in order
- Sometimes, a repetition of a part
- Assumed knowledge about context
- An expected result (yum!)

Additional feature of recipes

- And other common analogies, such as
  - Knitting patterns for knitting
  - Sheet music for playing music and

Additional feature of recipes

- And other common analogies, such as
  - Knitting patterns for knitting
  - Sheet music for playing music and
- If you go to a great restaurant, one with a fancy table in the kitchen where you can watch the chefs all night long, and you go there twice a week for 15 weeks, then:
  - You must make more money than professors! And

Additional feature of recipes

- And other common analogies, such as
  - Knitting patterns for knitting
  - Sheet music for playing music and
- If you go to a great restaurant, one with a fancy table in the kitchen where you can **watch the chefs all night long**, and you go there **twice a week for 15 weeks**, then:
  - You must make more money than professors! And
  - ***You won't learn how to cook!***

INTRODUCTION TO THE LAW COMPONENT

Why Law and Computer Science?

- Isn’t law about rules? About memorizing rules?
- Partly, but you won’t have to memorize legal rules.
- We will talk about the other thing law is about:
  - *Policies*: what kind of society should we live in?

That Belongs With Computer Science?

- It does.
  - The language of computation is the language through which computer science shapes society.
  - Lawyers do the same with talk about laws and legal concepts.
- Now more than ever they need to talk *to each other*.
- **And: even if you will not be lawyer or a computer scientist**, you need to understand and be part of that conversation.

One Reason

- “History will record what we, here in the early decades of the information age, did to foster freedom, liberty, and democracy. Did we build information technologies that protected people’s freedoms . . .? Or did we build technologies that could easily be modified to watch and control?”
  - Bruce Schneier

Another Reason

- It is important to seeing your life in a context that reveals the meaning and importance of your particular actions.
- An important part of the modern context is how computer science and law shape our world.

Can We Really Pull This Off?

- Yes.
- We are *not* offering what is really *two* courses: one part computer science, one part law.
- We are offering *one* course with those two *blended* together.
- That is how you learn to speak and think CS + law.

Law Topics

- You will learn to write code that
  - analyzes data,
  - Accesses online information.
- Your code illustrates our two main law topics:
  - Using computers to encroach on privacy.
  - Defending online data from unauthorized access.

An Example

- You will write a program (a web crawler) that extracts information from websites.
- Preparing for this course, I tested a web crawler that included this code:

```
connection = ur.urlopen(“some_website")
content = connection.read()
```

- I was careful about *what* websites.
- Why?

Code and Law

```
connection = ur.urlopen(“some_website")
content = connection.read()
```

CFAA

Copyright

Contracts

Contracts Because . . .

- "You will not . . . access Facebook, using automated means (such as harvesting bots, robots, spiders, or scrapers) without our prior permission." https://www.facebook.com/legal/terms
- In some jurisdictions, violating this contractual provision can open you to criminal liability under the Computer Fraud and Abuse Act.

Should Things Be That Way?

- The best answers will come from those who speak CS + Law.

BORING SYLLABUS STUFF

DON'T CHEAT: PENALTIES HARSH
DO READ THE SYLLABUS