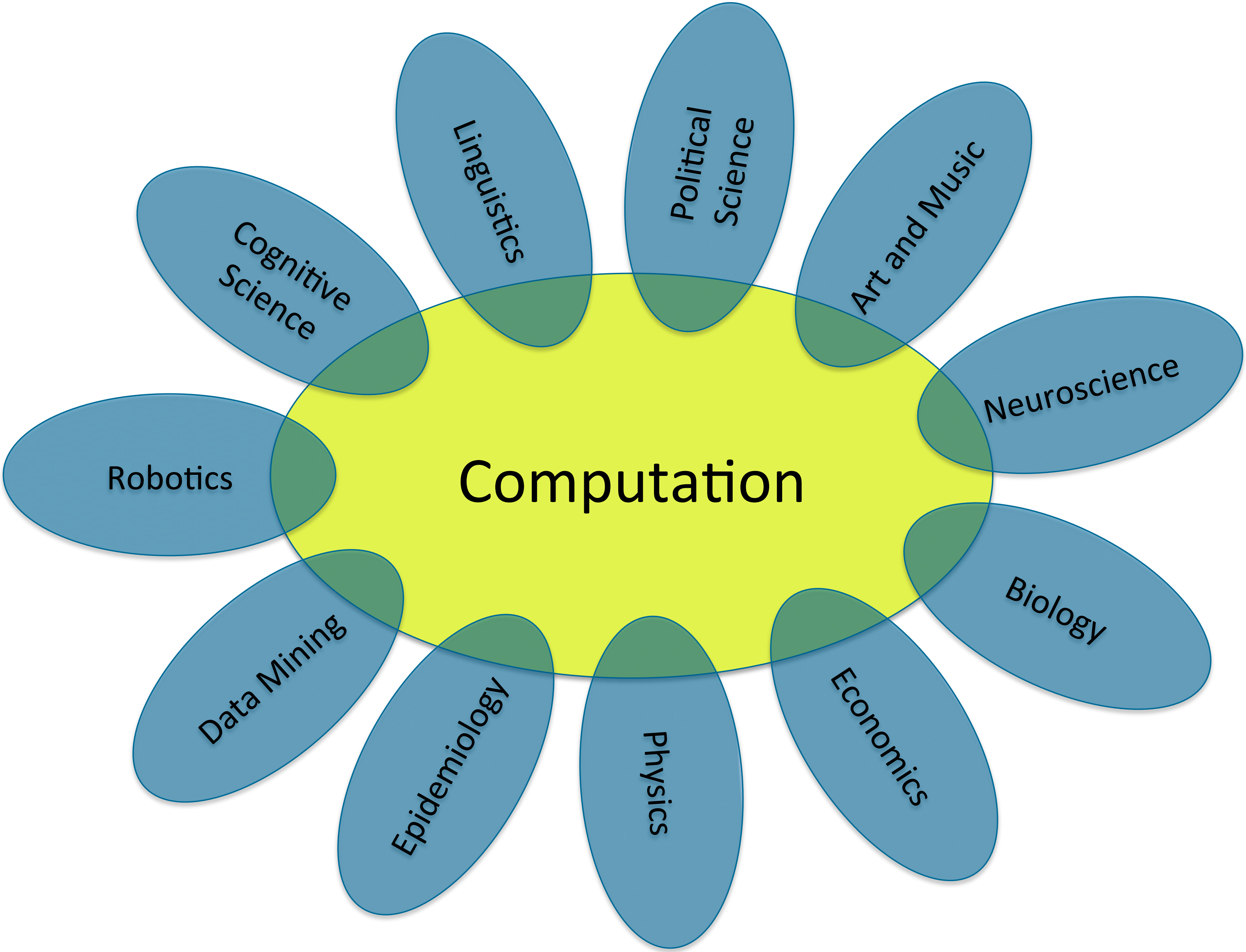
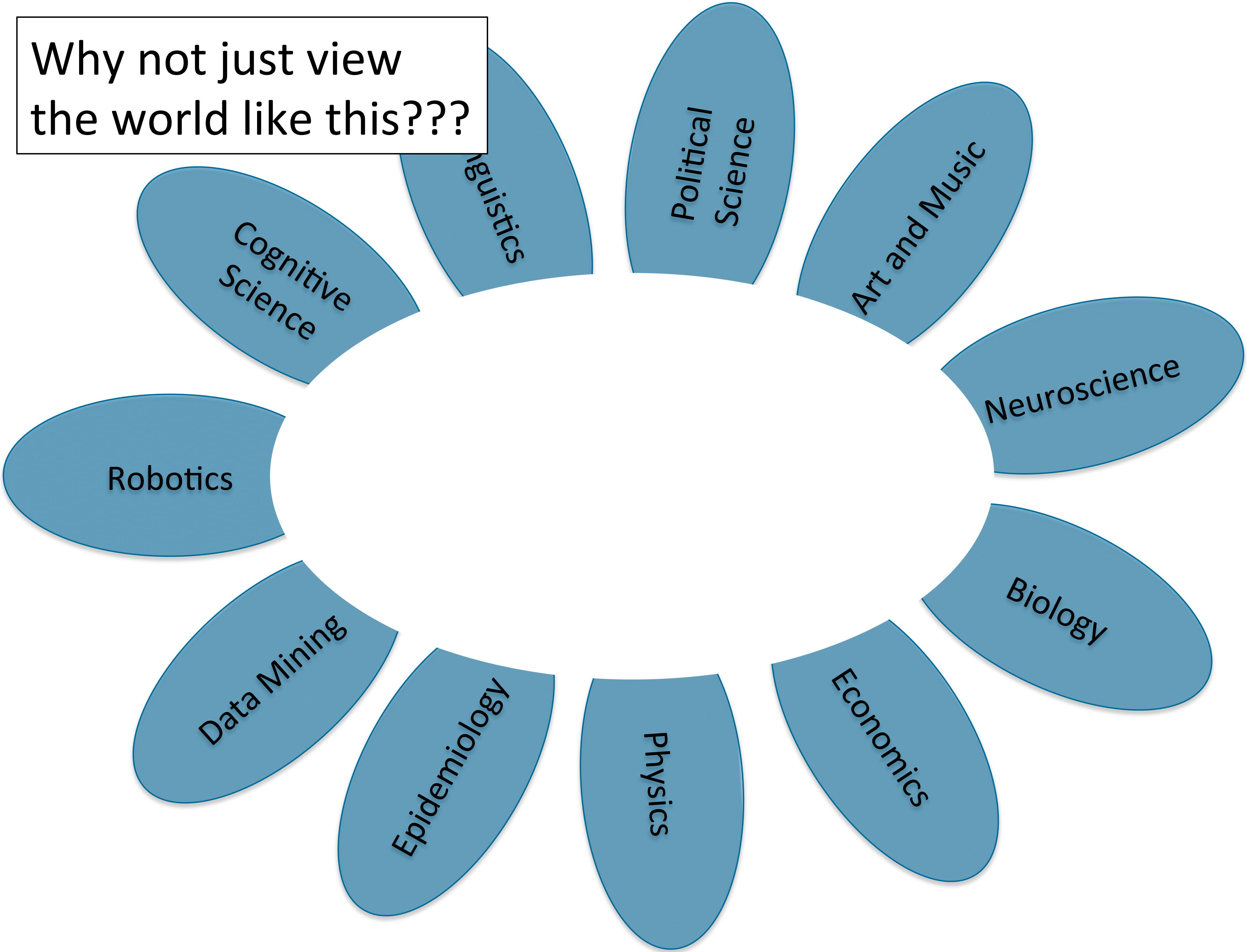
Software Design: A Look Back and Where to Next

Day 26


Computation
Linguistics
Political Science
Art and Music
Neuroscience
Biology
Economics
Physics
Epidemiology
Data Mining
Robotics
Cognitive Science

Why not just view the world like this???



What Makes Computing Such a Potent Skill? Or, **How this Class Will Give You Superpowers**[1].

If you know how to use professional software engineering tools and processes, you can:

- Combine multiple existing software packages to do something awesome
- Build your own tool from scratch to solve problems (or help some people) that are important to you

[1] The SoftDes faculty assume no liability if this class does not actually give you superpowers

What Makes Computing Such a Potent Skill? Or, **How this Class Will Give You Superpowers**[1].

If you know how to frame a problem computationally, you can:

- Develop creative solutions to hard problems
- See connections between your problem and other problems that initially seemed unrelated

[1] The SoftDes faculty assume no liability if this class does not actually give you superpowers

What Makes Computing Such a Potent Skill? Or, **How this Class Will Give You Superpowers**[1].

If you know how to effectively communicate and understand computational ideas, you can:

- Maximize the impact of code you write
- Work effectively (speak the same language) as software developers that you might collaborate with or hire

[1] The SoftDes faculty assume no liability if this class does not actually give you superpowers

We Hope:

1. that you feel you have acquired a superpower.
2. that if you never take another computing class, that you draw on what you learned in this class somewhere down the line.
3. that if you are take more computing classes, that we have set you up with a great foundation.
4. that you had a lot of fun.
5. that you help us improve the course.

Where to next?

YOU
ARE
HERE

SoftDes

Beyond Python?

- https://griffsgraphs.files.wordpress.com/2012/07/programming-paradigms_label2.png
- Focus on learning core concepts not flavor of the week technologies
- Also checkout Rosetta Code (implementations of the same algorithm in multiple languages)

Beyond Python

- Python is an interpreted language → relatively slow
- If you want faster…
  - Profile your code (we've seen this one)
  - Try Cython, or PyPy
  - Try different algorithms and / or data structures
  - Then possibly try a compiled language like Java, C, or C++

Beyond Python

- If you want to move to mobile: Objective C, Swift, and Java
- If you want to move to the web: Javascript

Computation in the Olin Curriculum

SoftDes is the entry point to most of the E:C curriculum

- Data Science
- SoftSys
- FOCS
- Mobile Proto
- OlinJS
- Computer Networks
- ***Recent electives:*** Visualizing Data, Video Game Design, Artificial Intelligence

Computation in the Olin Curriculum

- Comp Arch
- SCOPE
- POE
- Elecanisms
- SigSys
- Robotics (specifically CompRobo and Robotic Systems Integration)

Extracurricular Avenues

- SLAC
- Do-ML
- Computing Conversations
- **Mailing lists:** olin-cs Google group, devtalk

SoftDes

Other Avenues

(1) First, find some curious peers to learn with

(2) Decide on some materials / projects

- Go through a textbook (Allen's books are great choices)
- Read some research papers in an area you care about
- ACM Programming Competitions
- Project Euler
- Robotics: ROS (you don't even need a real robot if you use Gazebo)
- Computer Vision: learn more OpenCV
- Kaggle and DrivenData (learn more ML)
- Visualization: D3 (Javascript)
- Contribute to an open source project
- A really nice list - https://wiki.python.org/moin/ProblemSets

Do Something

- Keep working on your final project
- Expand on one of the toolboxes
- Build a library
- Bring a computational component to a project in another class
- Do some outreach
- Start a company