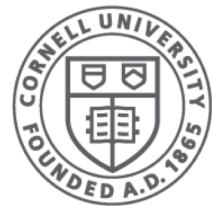


<http://www.cs.cornell.edu/courses/cs1110/2018sp>

Life after CS 1110

CS 1110

Introduction to Computing Using Python



Cornell CIS
COMPUTING AND INFORMATION SCIENCE

[E. Andersen, A. Bracy, D. Gries, L. Lee, S. Marschner, C. Van Loan, W. White]

Obvious Next Step: CS 2110

- **Programming in Java**

- Basic Java syntax
- Static vs. Dynamic Types
- Adv. Java Topics (e.g. Threads)

} Java Specific

} Language Independent

Major CS Topic

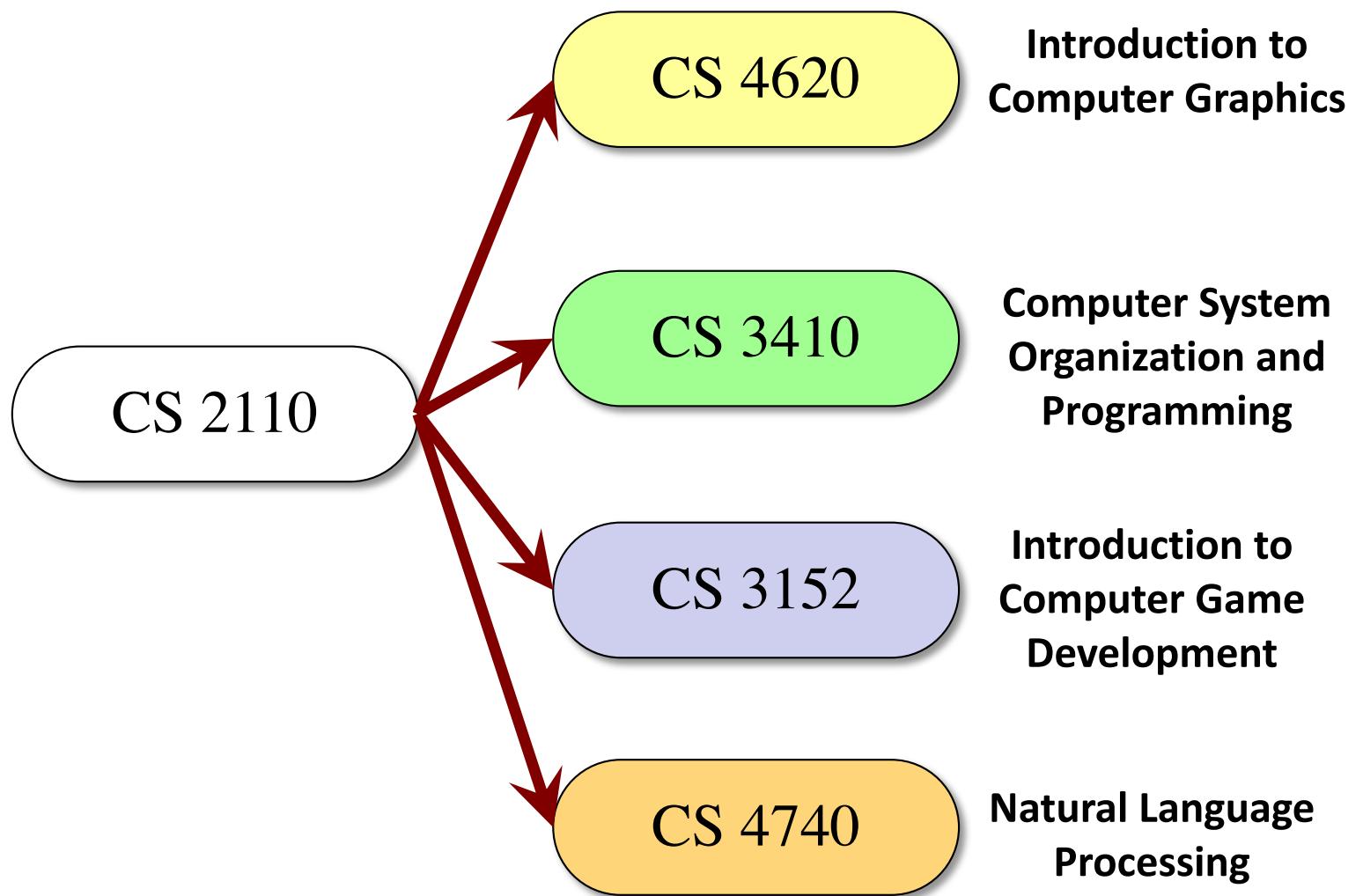
- **OO Theory**

- More design patterns
- Interface vs. Implementation

- **Data Structures**

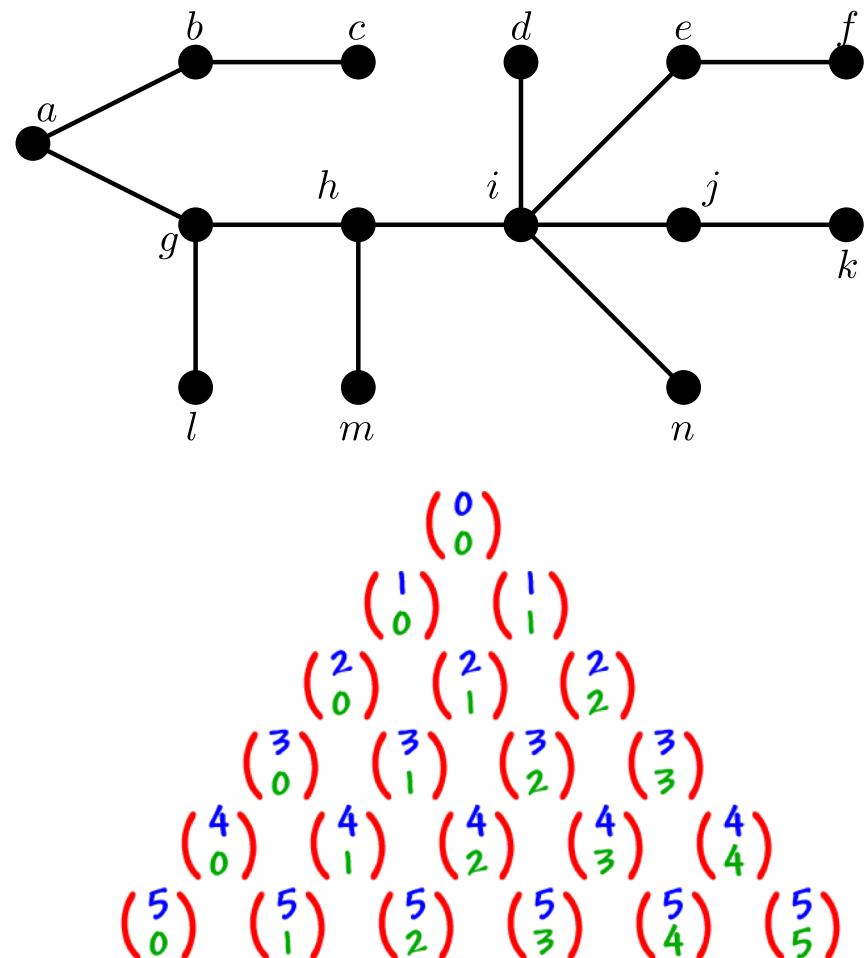
- Binary Trees
- Linked Lists
- Graphs

CS 2110 Immediately Opens your Options

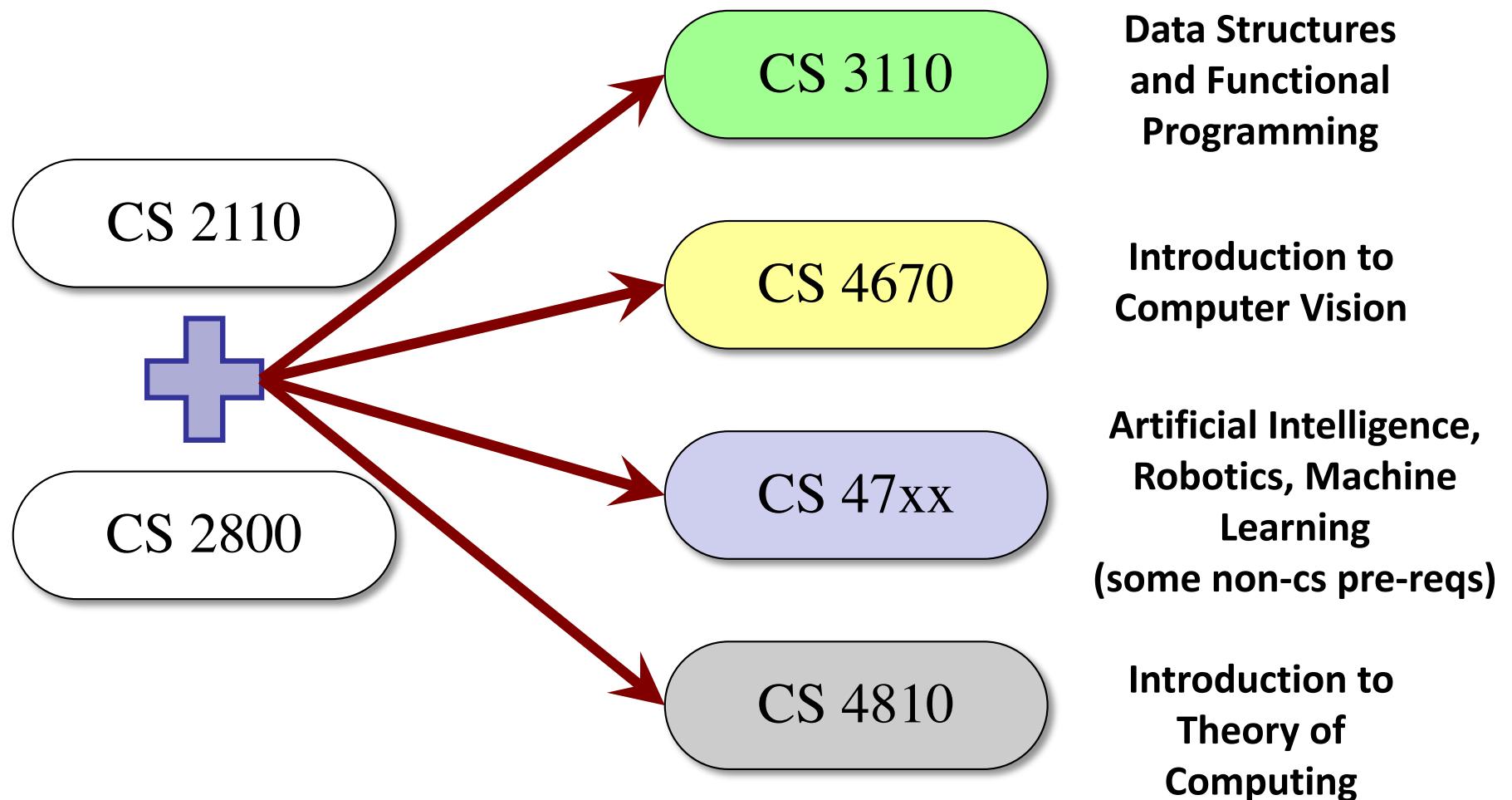


CS 2800: The Other Important Course

- CS requires a lot of math
 - Analyzing code performance
 - Analyzing data
 - Proving code correctness
- Calculus is “wrong math”
 - Data is rarely “continuous”
 - Limited to specific uses
(e.g. spatial data)
- “Grab-bag” course
 - All math needed for CS
 - Includes writing proofs



CS 2110 + CS 2880 = Even More Options



Computer Science Course Numbers

- Programming Languages **x1xx** (e.g. 1110, 2110)
- Scientific Computing **x2xx** (e.g. 4210)
- Data Management **x3xx** (e.g. 3300, 4320)
- Systems **x4xx** (e.g. 3410, 4410)
- Computational Biology **x5xx** (e.g. 5555)
- Graphics and Vision **x6xx** (e.g. 4620)
- Artificial Intelligence **x7xx** (e.g. 4758, 4700)
- Theory **x8xx** (e.g. 4810, 4820)
- Research **x9xx** (e.g. 4999)

Level Area

Computer Science Course Numbers

- Programming Languages **x1xx** (e.g. 1110, 2110)
- Scientific Computing **x2xx** (e.g. 4210)
- Data Management **x3xx** (e.g. 4320)
- Systems **x4xx** (e.g. 410)
- Computer Networks **x5xx** (e.g. 455)
- Graphics **x6xx** (e.g. 4620)
- Artificial Intelligence **x7xx** (e.g. 4758, 4700)
- Theory **x8xx** (e.g. 4810, 4820)
- Research **x9xx** (e.g. 4999)

Separation not perfect;
there is a lot of overlap

Level Area

Programming Languages

- **Adv. Language Topics**

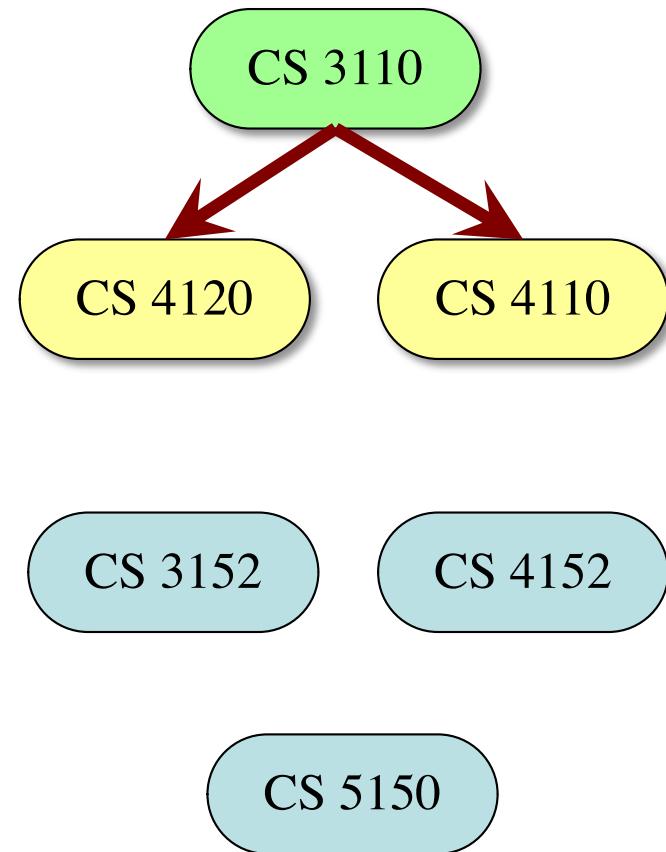
- Functional languages
- Streaming languages
- Parallel programming

- **Language Theory**

- New languages/compilers
- Software verification

- **Software Engineering**

- Design patterns
- Architecture principles



Scientific Computing

- **Calculus + Computing**

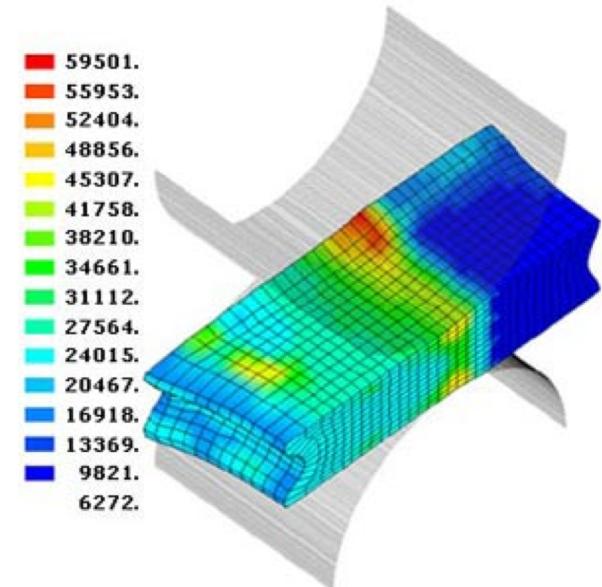
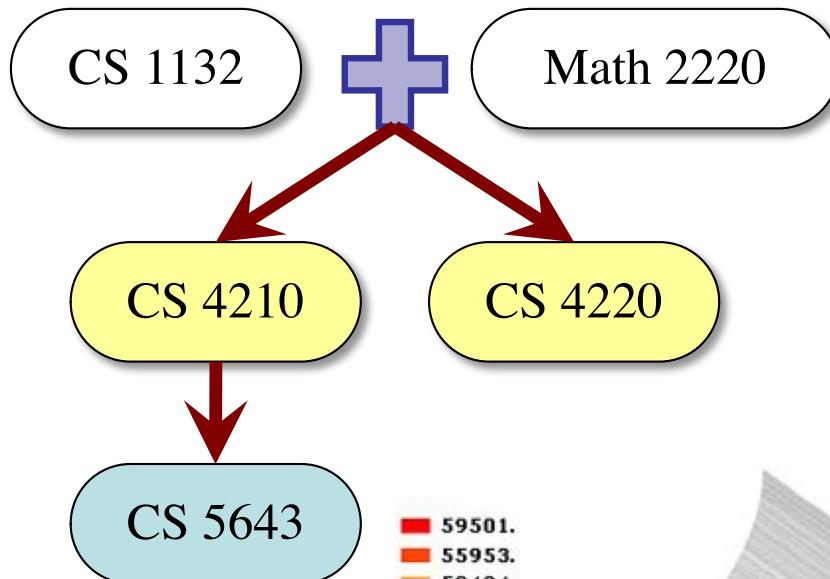
- Problems from other science domains
- Process with computer

- **Applications**

- Complex simulations
- Physics (games!)

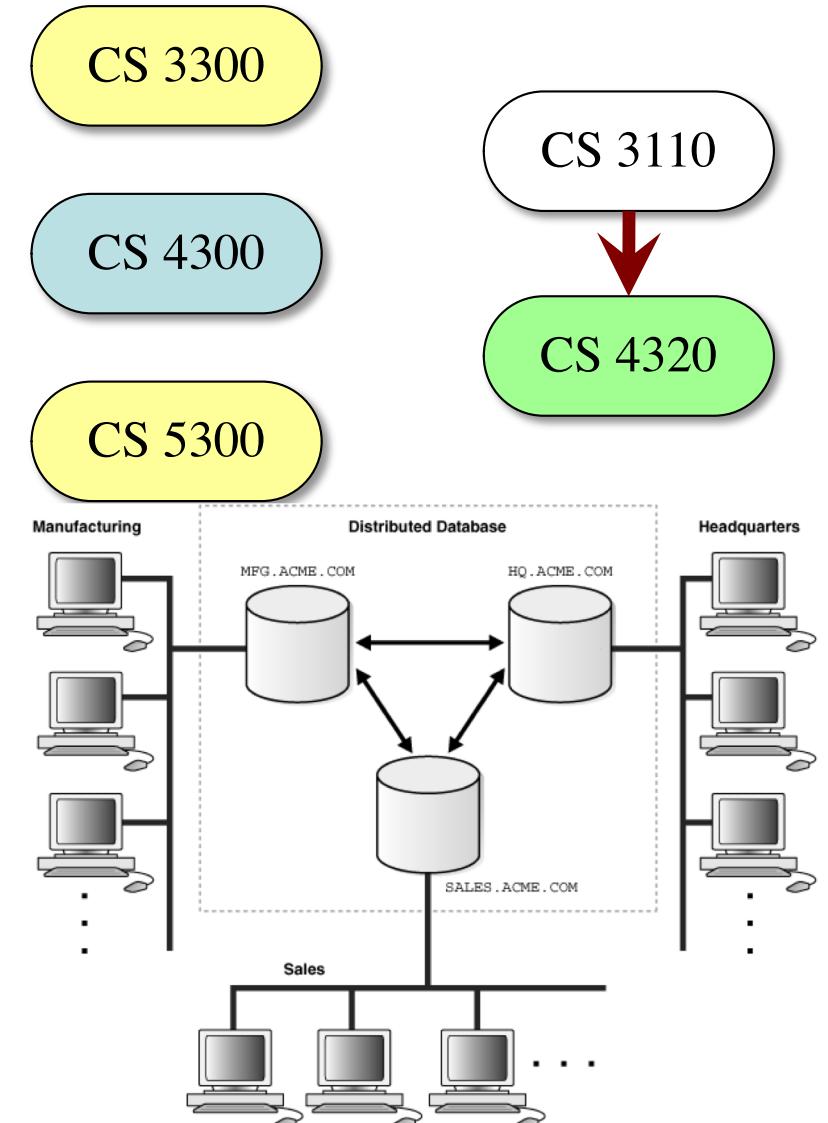
- **Challenge:** Performance

- Programs can run for days!
- How do we make faster?



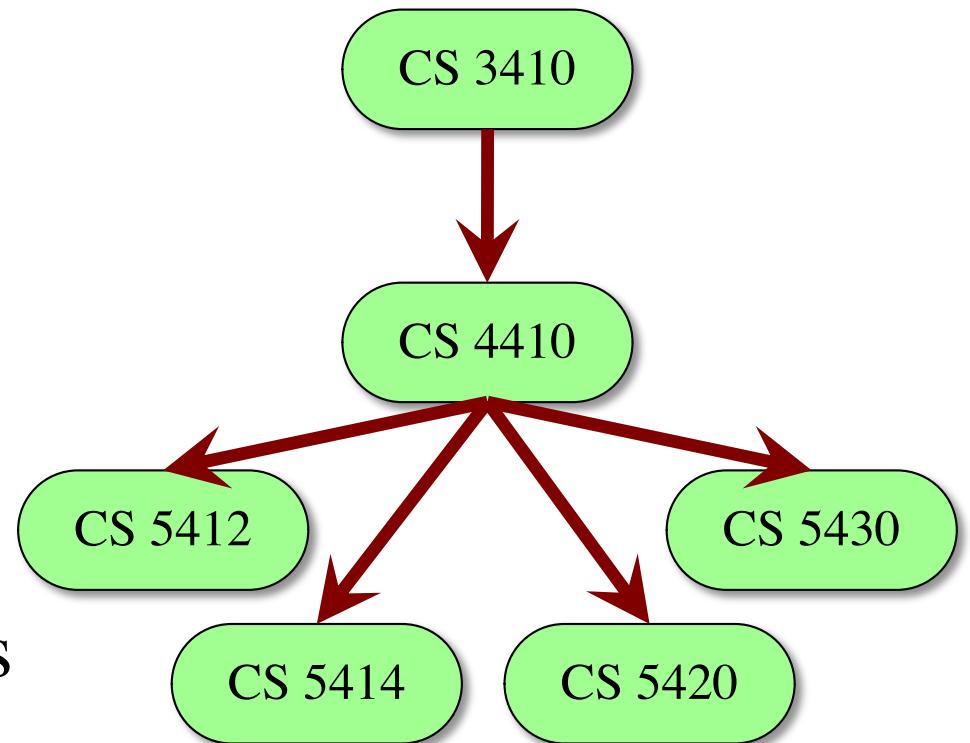
Data Management

- **Modern Web Apps**
 - Storing user/session data
 - Coordinating users
- **Databases**
 - Query languages
 - Database optimization
 - Organizing your data
- **Information Retrieval**
 - Searching
 - Data analysis



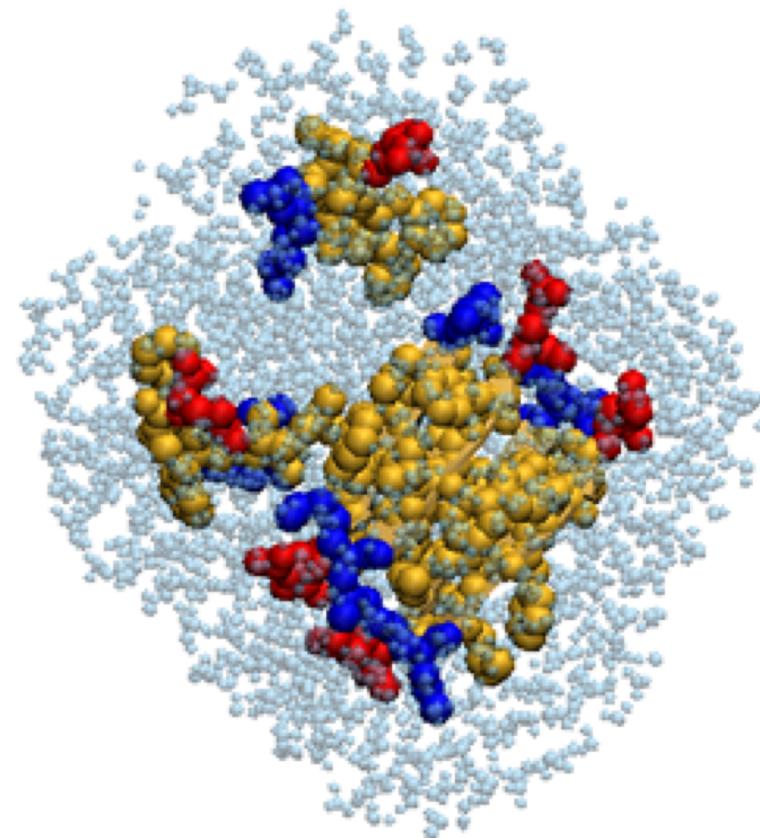
Systems

- **Building BIG software**
 - Operating systems
 - Distributed applications
(e.g. online, networked)
 - Cloud computing
- Also **System Security**
 - Though that is spread about
- Senior/masters level classes
 - Bulk of the 5xxx courses
 - But great project courses!



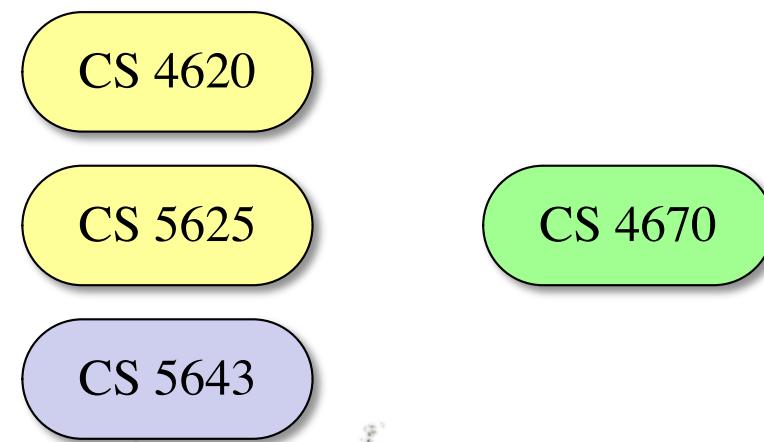
Computation Biology

- No undergrad classes
 - Too much to learn
 - Masters/PhD level
- Undergrad options
 - **BTRY 4840:**
Comp. Genomics
 - BSCB department
- Hoping to improve...



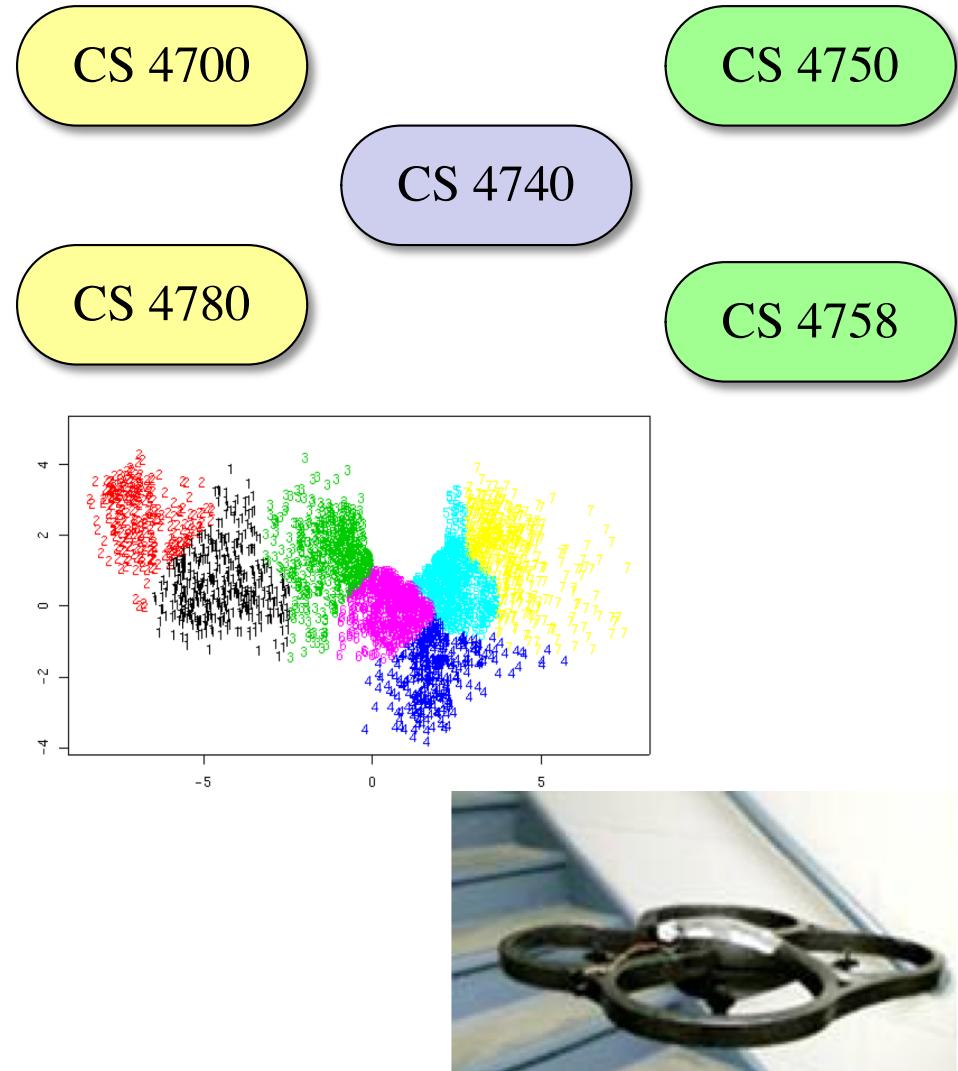
Graphics and Vision

- **Not** modeling/art!
- **Rendering & Animation**
 - Illumination/reflection
 - Cloth/hair simulation
 - Water and fluids
- **Processing Images**
 - Recognizing shapes
 - Assembling 3D models from 2D pictures
 - Smart cameras



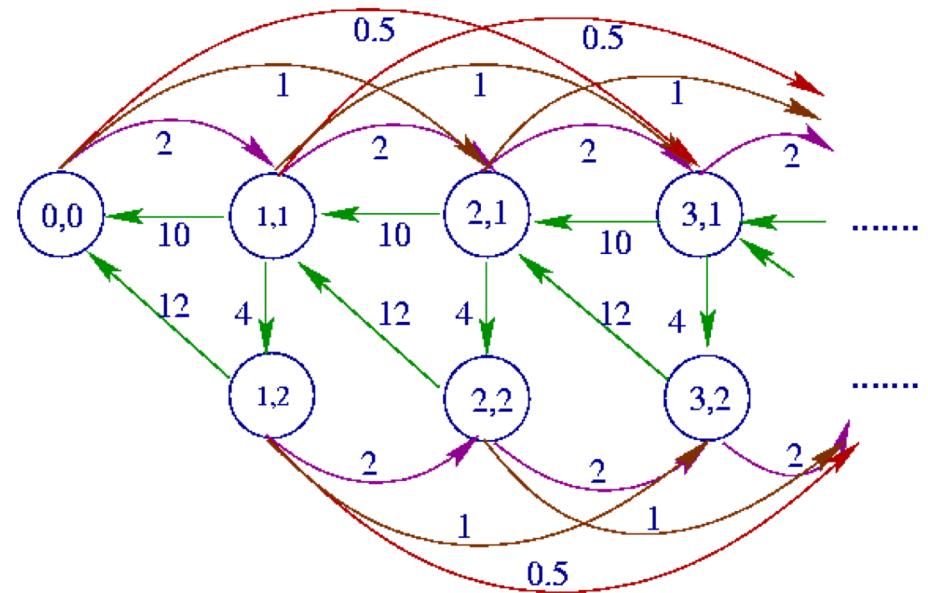
Artificial Intelligence

- **Not** sentient computers
- **Machine learning**
 - Discovering patterns
 - Making predictions
- **Natural Language Proc.**
 - Automatic translation
 - Searching text/books
 - Voice-control interfaces
- **Robotics**
 - Autonomous control



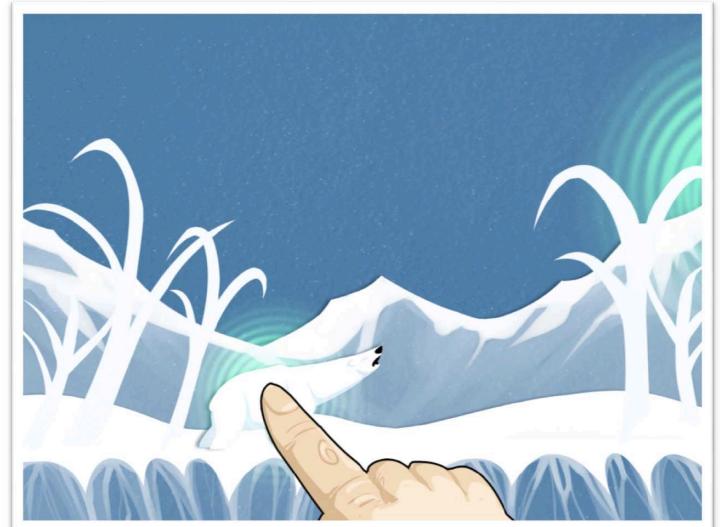
Theory

- **Analysis of Algorithms**
 - What is *possible*?
 - What is *feasible*?
- **Analysis of Structures**
 - Social network theory
 - Complex data structures
- **Cryptography**
 - Theory side of security
- Perhaps the most famous group in the department



What About Games?

- CS 3152, Spring only
 - Prereq: CS 2110
 - But CS 3110 a big help
- Build game from scratch
 - Want it to be innovative
 - You own the IP
- Interdisciplinary teams
 - 5 to 6 people on a team
 - With artists/designers
- **Final:** public showcase



Games and the Designer Track

- Coding not your thing?
- INFO 3152 (co-meets)
 - Artists/designer track
 - No formal training needed
 - Submit me a portfolio
- Recommend: INFO 2450
 - Start of the HCI sequence
 - How design effects the user experience
 - Fall course; no prereqs



Computer Science not your



?

Try one of our neighbors!

- Information Science
- Statistics
- Operations Research & Information Engineering
- Electrical and Computer Engineering



Cornell CIS
COMPUTING AND INFORMATION SCIENCE

InfoSci Classes you could have already taken

INFO 1300

**Introductory Design
and Programming for
the Web**

INFO 2040

Networks

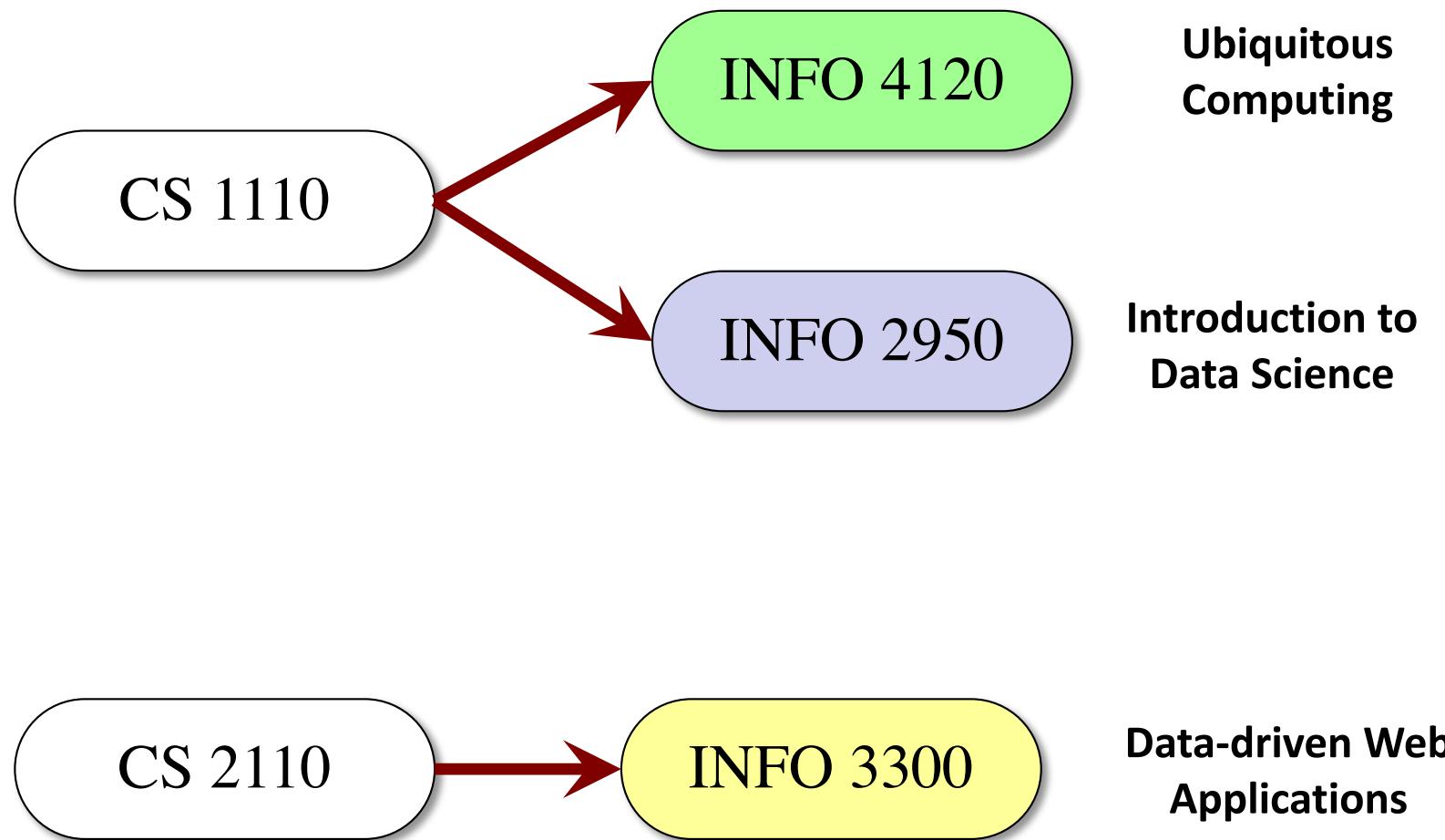
INFO 2770

**Excursions in
Computational
Sustainability**

INFO 3140

**Computational
Psychology**

InfoSci Classes you can take after some CS



Not a complete list!

Good Bye!