Obvious Next Step: CS 2110

- **Programming in Java**
  - Basic Java syntax
  - Static vs. Dynamic Types
  - Adv. Java Topics (e.g. Threads)
- **OO Theory**
  - More design patterns
  - Interface vs. Implementation
- **Data Structures**
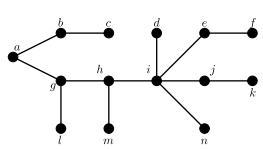 - Binary Trees
  - Linked Lists
  - Graphs

Java Specific

Language Independent

Major CS Topic

1

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**

$$\binom{0}{0}$$
$$\binom{1}{0}\ \binom{1}{1}$$
$$\binom{2}{0}\ \binom{2}{1}\ \binom{2}{2}$$
$$\binom{3}{0}\ \binom{3}{1}\ \binom{3}{2}\ \binom{3}{3}$$
$$\binom{4}{0}\ \binom{4}{1}\ \binom{4}{2}\ \binom{4}{3}\ \binom{4}{4}$$
$$\binom{5}{0}\ \binom{5}{1}\ \binom{5}{2}\ \binom{5}{3}\ \binom{5}{4}\ \binom{5}{5}$$
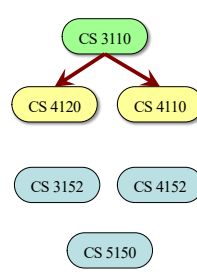
2

Higher Level Computer Science Courses

- 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)
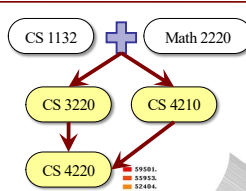- Research   x9xx   (e.g. 4999)

3

Programming Languages

- **Adv. Language Topics**
  - Functional languages
  - Streaming languages
  - Parallel programming
- **Language Theory**
  - New languages/compilers
  - Software verification
- **Software Engineering**
  - Design patterns
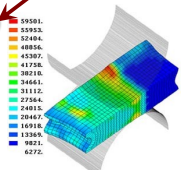 - Architecture principles

CS 3110
CS 4120   CS 4110
CS 3152   CS 4152
CS 5150

4

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?

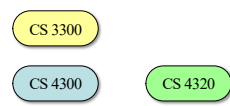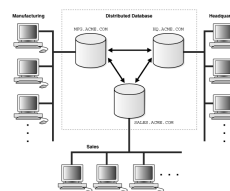CS 1132   +   Math 2220
CS 3220   CS 4210
CS 4220

5

Data Management

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

CS 3300
CS 4300   CS 4320

6

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!

CS 3410 → CS 4410 → CS 5412, CS 5430, CS 5414, CS 5420

7

Computation Health/Biology

- **No undergrad classes**
  - Used at CornellTech
  - Too much to learn
- Once hoped for Ithaca
  - But hard to hire in CS
  - Faculty better fit for Bio
- Now in Comp. Bio dept.
  - Separate department
  - But part of CIS school
  - Also a graduate program
  - But has concentration in Bio

8

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

CS 4620, CS 5625, CS 4670, CS 5643

9

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

CS 4700, CS 4740, CS 4750, CS 4780, CS 4744, CS 4754, CS 4783, CS 4745, CS 4758, CS 4786, CS 478x

10

Theory

- **Analysis of Algorithms**
  - What is *possible*?
  - What is *feasible*?
- **Analysis of Structures**
  - Social network theory
  - Complex data structures
- **Cryptography**
  - Theory side of security
- Area responsible for founding dept. in 1965

CS 4810, CS 4830, CS 4860, CS 4820

11

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
  - 7 to 8 people on a team
  - With artists/designers
- **Final**: public showcase

12