Overview of the New Undergraduate Computer Science Curriculum

Mehran Sahami

CS298 — September 23, 2008

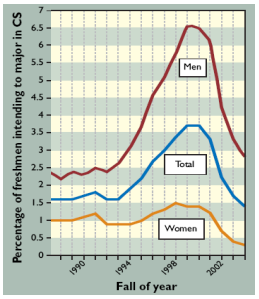
The Most Famous Chart in CS Education

Source: *Higher Education Research Institute*, UCLA

Publicized by David Patterson in CACM, Sept. 2005

By 1999, everyone and their dog wanted to major in CS

Graphic thanks to Andy Maag



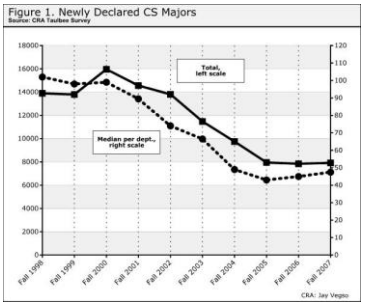
Outline

- The Computer Science student pipeline
- Misperceptions of the field
- CS curriculum revision at Stanford
  - Structure
  - Details
  - Issues
- Future considerations

Outline

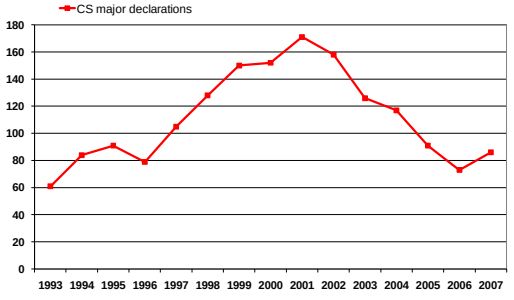
- The Computer Science student pipeline
- Misperceptions of the field
- CS curriculum revision at Stanford
  - Structure
  - Details
  - Issues
- Future considerations

CRA Taulbee Survey on CS Declarations

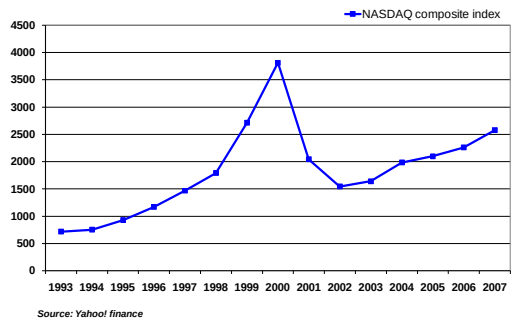


Source: **Jay Vegso,** ***Enrollments and Degree Production at US CS Departments Drop Further in 2006/2007,*** **March 1, 2008** ***(from CRA website)***

The Data From Stanford



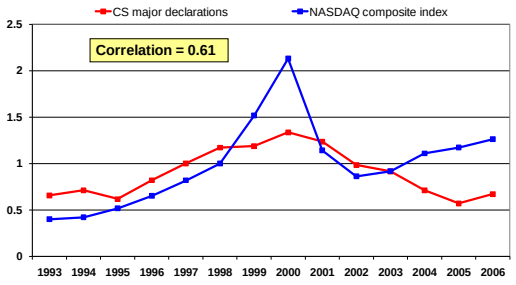
A Slightly More Well Known Graph



Source: Yahoo! finance

The Obvious Correlation

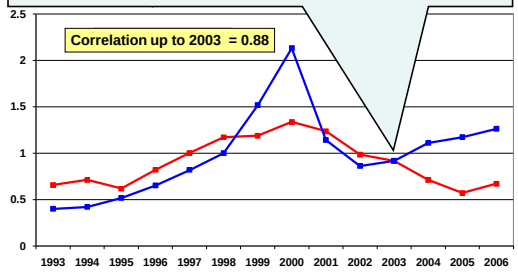
- Normalize both graphs by 1998 values
  - Adjust for a one year lag time in declarations



What Happened in 2003?

By 2003, … sensational news stories appeared about a supposedly horrific loss of these [computer programming] jobs [due to offshoring].

-- The Washington Times, June 6, 2004



Outline

- The Computer Science student pipeline
- Misperceptions of the field
- CS curriculum revision at Stanford
  - Structure
  - Details
  - Issues
- Future considerations

Pop Quiz!

According to the Bureau of Labor Statistics…

1. Computer programmers write, test, and maintain the detailed instructions that computers follow.

2. Software engineers design, create, and modify computer applications and systems.

Select a choice for each blank above:

A. Software engineers

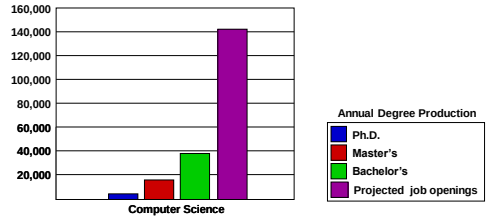
B. Computer programmers

The Truth On Offshoring

- Confusion at the Bureau of Labor Statistics
  - Projected Job Growth from 2006 to 2016
    - "Computer programmer": ***below*** average
    - "Computer scientists" & "software engineers": ***above*** average
- More IT jobs today in US than during boom
  - Despite significant increase in offshoring over past 5 years

  Source: *Globalization and Offshoring of Software: A Report of the ACM Job Migration Task Force* (citing the Bureau of Labor Statistics), 2006.
- Need to create awareness of "CS in the large"
  - CS is increasingly fundamental to work in other fields

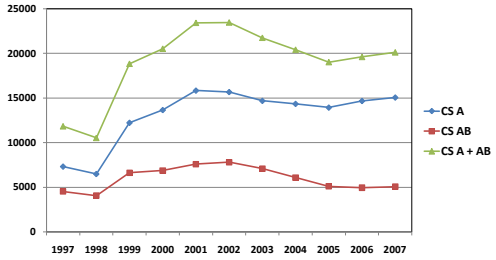
Supply Versus Demand



Sources: Courtesy of Eric Roberts -- adapted from a presentation by John Sargent, Senior Policy Analyst, Department of Commerce, at the CRA Computing Research Summit, February 23, 2004. Original sources listed as National Science Foundation/Division of Science Resources Statistics; degree data from Department of Education/National Center for Education Statistics: Integrated Postsecondary Education Data System Completions Survey; and NSF/SRS; Survey of Earned Doctorates; and Projected Annual Average Job Openings derived from Department of Commerce (Office of Technology Policy) analysis of Bureau of Labor Statistics 2002-2012 projections. See http://www.cra.org/govaffairs/content.php?cid=22.

Entrants Into CS Pipeline Not Increasing

- AP exams as a leading indicator of interest
  - Note: AP CS exam switched to Java in 2004



Source: College Board

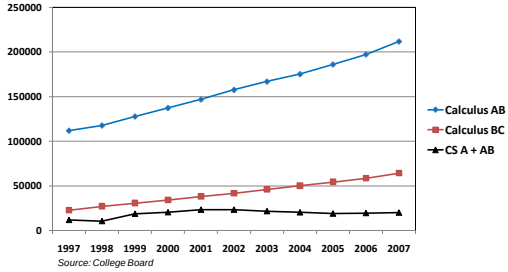
College Board Dropping CS AB Exam

- After 2008-09, College Board is dropping CS AB exam
  - Along with *Italian*, *Latin Literature* and *French Literature*

Despite the company, I don't think Java can be considered a Romance language

Declines Not Seen in Any Other AP Exam

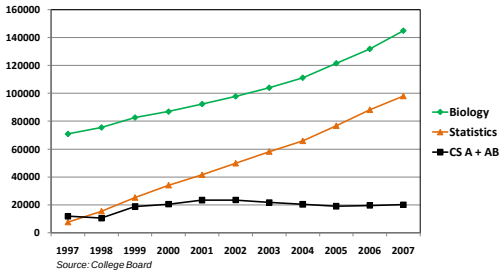
- Don't get too excited about up-tick in CS in 2006-07
  - But there is still healthy pipeline of technically-oriented students



Source: College Board

Non-Traditional Sources of CS Students

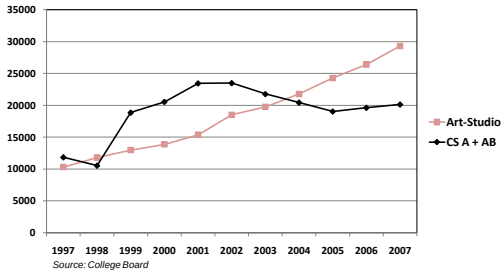
- Growth in disciplines which are increasingly CS-related



Source: College Board

Sobering Thought: Studio Art AP Exams

- *They have AP exams in Studio Art?*
  - *Yes, and they are taken more often than CS A & AB combined!*



Source: College Board

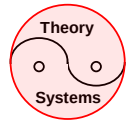
Outline

- The Computer Science student pipeline
- Misperceptions of the field
- CS curriculum revision at Stanford
  - Structure
  - Details
  - Issues
- Future considerations

Revised Curricular Structure: *Core*

Theory Core: 3 Courses

- Incorporates majority of current cs103a/b course topics
- Eliminates redundancies in existing courses
- CS-owned probability course with AI applications



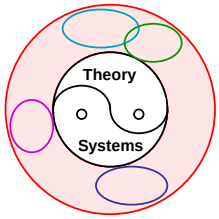
Systems Core: 3 Courses

- Incorporates much of current intro programming sequence
- Incorporates systems concepts in later programming projects
- CS106A considered "funnel" into core (not part of core)

Revised Curricular Structure: *Tracks*

~4 Courses

- Students must complete requirements for any one track
- Developing depth in a specialization
- Provide course options within each track
- Provide multi-disciplinary options
- Modularize curriculum



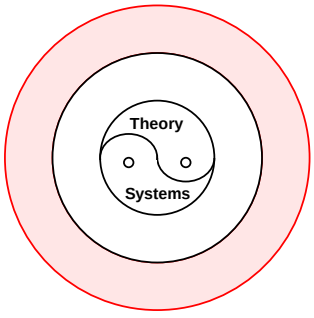
Why Tracks?

- Explicitly shows available options
  - Diversity of areas within computer science
  - Allows focus on areas of greatest interest
- Helps eliminate image of CS as "just programming"
  - Programming is the *means*, not the *ends*
  - Significant role of computing in inter-disciplinary work
  - Provides more context for what is possible with CS degree
  - Still, should not discount the importance of rigorous software engineering skills
    - *Don't "water down" the curriculum to just attract more students!*
- Provides organizational infrastructure
  - Easier to evolve major as the field evolves
  - E.g., add/drop/modify tracks (or programs in them)

Revised Curricular Structure: *Electives*

~2-4 Courses

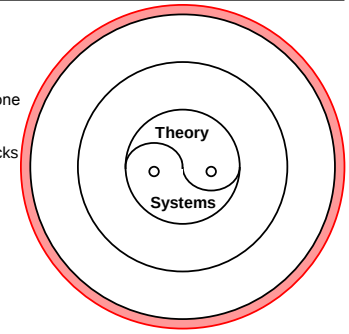
- Restricted electives
- Allow pursuing breadth and/or additional depth
- Track-specific elective options allow for interdisciplinary work



Revised Curricular Structure: *Capstone*

1 Course

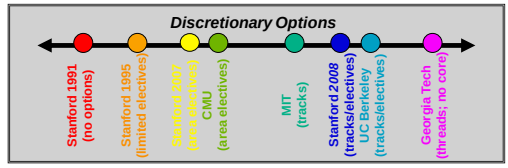
- "Senior project" capstone course
- Developing capstone courses to parallel tracks
- Both application development and research options



Structure Aligns With Broader Context

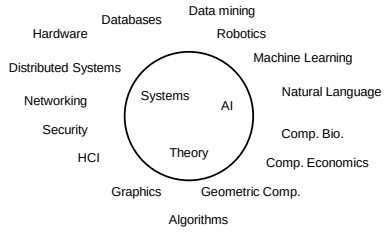
- IEEE-ACM Computing Curricula 2001 Report
  - Supports tracks model
  - Revision committee adopted modular structure to support adaptability
- ICER: Integrative Computing Education & Research
  - Change the popular image of computing
  - Encourage curricular experimentation and innovation
  - Make sure introductory students recognize that the field offers many opportunities
  - Strengthen interdisciplinary connections

Peer Institutions



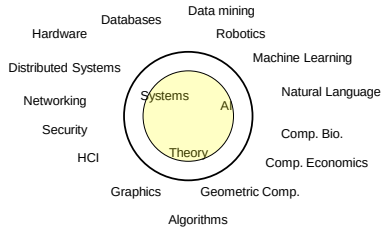
- Stanford's curriculum has evolved over time
  - The latest development is the largest shift
- Other institutions also moving toward more flexible models
- Several other institutions are currently examining the new curriculum at Stanford

Increasing the "Footprint" of CS



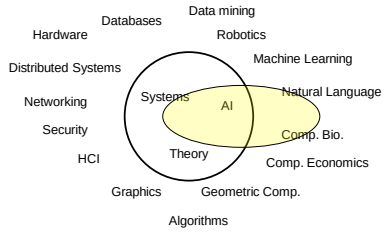
Editor's Note: Two-dimensional projection clearly does not capture the relative importance or organizational nuances of the field. Some topics may be closer to you than they appear on this slide.

"Footprint" of CS Students See Today



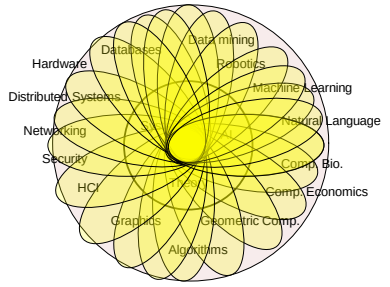
Editor's Note: Two-dimensional projection clearly does not capture the relative importance or organizational nuances of the field. Some topics may be closer to you than they appear on this slide.

Tracks Allow More Depth...

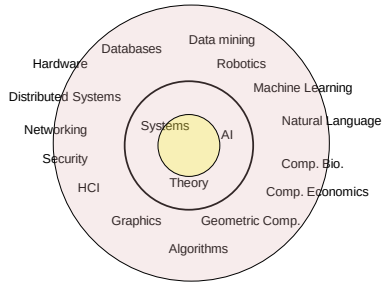


Total amount of material covered must remain the same

...in a More Diverse Set of Areas



Total Potential "Footprint" is Larger



Core material everyone sees is streamlined to accommodate

Vast Majority of CS Department Involved

Curriculum Committee

- Jerry Cain
- Bill Dally
- Vladlen Koltun
- Phil Levis
- John Mitchell
- Andrew Ng
- Nick Parlante
- Eric Roberts
- Mendel Rosenblum
- Mehran Sahami (Chair)
- Julie Zelenski

Beyond the Committee

- Alex Aiken
- Serafim Batzoglou
- Gill Bejerano
- David Dill
- Ron Fedkiw
- Hector Garcia-Molina
- Leo Guibas
- Pat Hanrahan
- Scott Klemmer
- Daphne Koller
- David Koslow
- Jean-Claude Latombe
- Marc Levoy
- Chris Manning
- David Mazieras
- Rajeev Motwani
- Serge Plotkin
- Bob Plummer
- Vaughan Pratt
- Tim Roughgarden
- Claire Stager
- Sebastian Thrun
- Jennifer Widom
- Terry Winograd
- Patrick Young
- Russ Altman
- Many additional faculty (email/informal meetings)

Shows real dedication to undergraduate education

Outline

- The Computer Science student pipeline
- Misperceptions of the field
- CS curriculum revision at Stanford
  - Structure
  - Details
  - Issues
- Future considerations

Discussion:
Why did you pursue CS (or not)?

Outline

- The Computer Science student pipeline
- Misperceptions of the field
- CS curriculum revision at Stanford
  - Structure
  - Details
  - Issues
- Future considerations

Systems I: Programming Abstractions

- **CS106B/X**
- Philosophy: *problem solving, basic abstract data types, and recursion*
- General Topics
  - Programming methodology (engineering, modularity, documentation)
  - Algorithmic thinking and problem solving
  - Data abstractions
    - Stacks
    - Queues
    - Linked lists
    - Hash tables
    - Binary trees
    - Generics/templates
  - Recursion
    - Procedural recursion
    - Recursive backtracking
  - Searching and sorting
  - Basic algorithm analysis (big-Oh) and comparison

Systems II: Computer Organization and Systems

- **CS107**, modified with material from CS143 & EE108B
- Philosophy: *From hardware up to the source code*
- General Topics
  - Machine architecture
    - Registers, ALU, CPU, RAM, I/O, basic assembly language
    - Caching, pipelining
  - Memory model
    - Pointers, Heap management, garbage collection
    - Low-level polymorphism and runtime type identification
    - Data representation
    - Facility with C programming as part of topical coverage
  - Compilation
    - Function call mechanics and stack frames
    - Semantic analysis
    - Simple (intermediate) code generation
  - Basic concurrency usage
    - Threading
    - Synchronization, locks and semaphores

Systems III: Principles of Computer Systems

- **CS110** – New course (not replacement for CS140 or CS244A)
- Philosophy: *Building larger scale systems using OS and networking abstractions*
- General Topics
  - Processes
    - Concurrency mechanics on a single processor
    - Context switching, interrupts and exceptions
    - Forking processes, process mechanics and management
    - Interprocess communication
    - Threading
  - Storage and file management
    - File systems
    - Virtual memory and paging
  - Networking
    - Sockets
    - Blocking vs. non-blocking strategies
    - Transport layer: TCP/IP
    - Network layer: names, routing
  - Understanding of distributed systems

Theory I: Mathematical Foundations of Computing

- **CS103** – New course leveraging current CS103A/B/X courses and CS154 (does not replace CS154)
- Philosophy: *Mathematical essentials for CS, with proofs*
- General Topics
  - Logic and proof techniques
    - Prop. and predicate logic (with quantification), formal proof methods
    - Applications: Satisfiability, SAT solving (Putnam-Davis)
  - Induction
    - Formal proofs and applications: program proofs, structural induction
  - Sets, functions, and relations
    - Theory and applications (error-correcting codes, social networks)
  - Intro to formal languages
  - DFAs, NFAs, and Regular Expressions
  - Context-free Grammars
  - Turing machines
    - TMs, TM program, Undecidability and the Halting problem
  - NP-completeness
    - P and NP, examples of NP-complete problems and reductions
    - SAT revisited and Cook's theorem

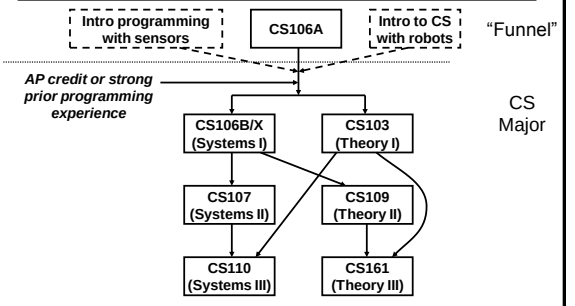
Theory II: Intro. to Probability for Computer Scientists

- **CS109** – New course
  - Replaces Stat116/MS&E120, adds CS applications
- Philosophy: *Probability relevant to CS, with applications*
- General Topics
  - Counting and Combinatorics
    - Combinations, Permutations, Pigeonhole principle
  - Probability theory
    - Random variables and event spaces
    - Conditional probability, independence, conditional independence
    - Distributions: Uniform, Binomial, Multinomial, Normal, Poisson
    - Point estimation, expectation, variance
    - Bayes' Theorem, Law of large numbers, Central Limit Theorem
    - Hypothesis testing
    - Applications: hashing, PageRank, data analysis, inference
  - Intro. to Machine Learning
    - Hypothesis spaces, learning as search
    - Data fitting, Naïve Bayes, Logistic Regression
    - Applications: Email spam filtering, Recommender systems

Theory III: Data Structures and Algorithms

- **CS161** – Modified with some differences in coverage
- Philosophy: *Analysis of data structures and algorithms*
- General Topics
  - Algorithmic complexity and analysis
    - Asymptotics: Big Oh, Omega, and Theta notation
    - Recurrence relations
    - Master theorem
  - Randomization, divide and conquer
    - Introduction to randomized algorithms
    - Quicksort, divide and conquer
  - Heaps and counting sort
  - Hashing
  - Tree and graph definitions and properties, BSTs
  - Greedy Algorithms (including min-cost spanning trees)
  - Dynamic programming
  - Graph algorithms, shortest paths, and applications
  - Blind and heuristic search (A*) in graphs

Prerequisite Structure



Courses at appropriate level of difficulty to mature from introductory to depth material in a reasonable manner

Summary of Changes

- Current CS103 sequence and STAT116/MS&E120 are replaced as Math requirements with Theory I and II
- CS161 incorporated into Theory III (remains CS depth course)
- CS154, CS121/221 (and other application course) not in core
- No change in Math units, 10 net units opened in CS depth
- Systems I considered an Engineering Fundamental
- Systems II and III are CS depth courses
- CS108, EE108B and Systems electives not in core
- 10 net units opened in CS depth
- Existing electives provide 6 additional units of CS depth
- **Net result**: 26 units opened in CS depth

Initial Set of Tracks Areas

- Artificial Intelligence
- Theory
- Systems
- Human-Computer Interaction
- Graphics
- Information
  - Management and applications of (un)structured information/data
- Biocomputation
- Unspecialized
  - Essentially, the current CS program
- *Individually Designed*

Track Structure

- Combination of track requirements and electives satisfies:
  - minimum of 7 courses AND minimum of 26 units
- All tracks have at least 4 (possibly more) required courses
  - Will generally leave room for 2 to 4 elective courses
  - Required senior project is not considered part of track
- Elective courses
  - Set of general CS electives that all students may choose from
  - Additionally, each track specifies *track-specific electives* that may count as elective courses only by students in that track
  - Track-specific electives allow for additional depth or related inter-disciplinary course options

Track General Form

- Requirements
  a) One or two "gateway" courses
  b) One or two courses from a menu of highly-related courses
  c) Selection of courses from (b) and/or from a list of more broadly related courses
- Track electives
  - Additional courses available to students in this track
  - Includes non-CS courses
  - Also includes (more) graduate courses in track area
  - All tracks include "General CS electives" in elective list

Artificial Intelligence Track

- Requirements
  a) CS221
  b) Any two of:
  CS223A, CS223B, CS224M, CS224N, CS226, CS227, CS228, CS229
  c) One additional class from category (b) or from the following:
  CS205A, CS222, CS224S, CS224U, CS225A, CS225B, CS227B, CS262, CS276, CS277, CS279, CS321, CS326A, CS327A, CS329, CS374, CS379, EE263, EE376A, Eng205, Eng209A, Ling180, MS&E251, MS&E339, MS&E351, Stat315A, Stat315B
- Track electives
  - Courses in categories (b) and (c) above, as well as:
  CS270, CS273A, CS274, CS275, CS278, EE364A, EE364B, Econ286, MS&E252, MS&E352, MS&E355, Phil151*, Phil152, Psych202, Psych204A, Psych204B, Stat200, Stat202, Stat205
  *students may not count both Phil151 and CS157 toward requirements.

Systems Track

- Requirements
  a) CS140
  b) One of: CS143 or EE108B
  c) Two additional courses from category (b) or from the following:
  CS144, CS145, CS155, CS240, CS240C, CS240D, CS242, CS243, CS244, CS245, EE271, EE282
- Track electives
  - Courses in category (c) above, as well as:
  CS240E, CS240X, CS244C, CS244E, CS315A, CS315B, CS343, CS344, CS344E, CS345, CS346, CS347, CS349*, CS448, EE382A, EE382C, EE384A, EE384B, EE384C, EE384S, EE384X, EE384Y
  *requires approval of undergraduate advisor.

Human-Computer Interaction Track

- Requirements
  a) CS147, CS247 *(HCI Foundations)*
  b) Any one of: CS148, CS248, CS376, CS377, CS378 *(Advanced HCI)*
  c) Any one of: CS108, CS140, CS221, CS223B, CS229, CS249A *(Buttressing CS)*
  d) Any one of: Psych55, Psych252, MS&E184, ME101, ME115 *(Designing for People)*
- Track electives
  – Courses in categories (b), (c), and (d) above, as well as:
  ArtStudi60, Comm269, CME340, CS447*, CS448B*, Ling180, EE118, MS&E216A, Psych205, Psych221
  *requires approval of undergraduate advisor.

Biocomputation Track

- Requirements (CS oriented version of BMC Informatics Track)
  – Mathematics: (1 course less than CS)
    • Math 41, Math 42, Theory I, Theory II
    • One of: Stat141, Stat203, Stat205, Stat215, Stat225
  – Science: (2-3 courses more than CS)
    • Physics 41, Chem 31A/B or 31X, Chem 33
    • Bio41, 42 or HumBio2A, 3A,
  – Engineering Fundamentals: (1 course less than CS)
    • CS106B/X (Systems I)
    • Elective
  – Additional CS Core: (same as CS)
    • Theory III, Systems II, Systems III
  – Biocomputation Track: (10-11 units)
    a) Any one of: CS121, CS221, CS228, CS229, CS223B
    b) Any one of: CS270, CS273A, CS274, CS275, CS278, CS279, CS262
    c) One additional course from (a) or (b) or :CS145, CS147, CS148 or CS248
  – Four Biocomputation Electives: (12-13 units; different than CS)
    • One course from either general CS of BMC Informatics elective lists
    • One course from BMC Informatics elective list
    • One course from either BMC Informatics, Cell/Mol, or Organs elective lists
    • One course from either BMC Cell/Mol or Organs elective lists
  – **Total**: 93-99 units versus 93 units for Standard CS Track

Unspecialized Track

- Requirements
  a) CS154
  b) Any one of: CS140, CS143
  c) One additional class from category (b) or from the following: EE108B, CS144, CS155, CS240D, CS242, CS244
  d) Any one of: CS121 or CS221, CS223A, CS223B, CS228, CS229
  e) Any one of: CS145, CS147, CS148 or CS248, CS262
- This is basically our current curriculum
  – Adapted to fit into new track structure
  – Some additional options for AI courses
- Allows "in-flight" students to graduate under new program

Individually Designed Track

- Students may propose Individually Designed Track
- Must be an intellectually coherent program of study
  – Should justify program and why it cannot be satisfied via an existing track
- Must specify equivalent of track and electives
  – Min. of 7 courses; at least 4 must be CS courses numbered 100 or above
    • Each course must be taken for a minimum of 3 units
  – Minimum of 26 total units for track + electives
- Must be approved by advisor and Associate Chair
  – Approvals obtained at least 2 quarters prior to completion of program
- Cannot modify any non-track/elective requirements
  – E.g., SoE requirements (Math, Science, Eng Fundamentals) cannot be modified
  – Must include all CS Core courses

Unit Calculations

- Core (29 units)
  – Theory: 2 courses @ 5 units, 1 course @ 4 units = 14 units
  – Systems: 3 courses @ 5 units = 15 units
- Upper division (29 units)
  – Track: 4-5 courses
  – Electives: 2-4 courses
  } Minimum of 7 courses (at least 26 units)
  – Capstone: 3 units
- Total related units = 58 units (same as before)
  – 10 units are Mathematics (same as before)
  – 5 units are Eng. Fundamentals (same as before)
- Total CS depth units = 43 units (same as before)

Curriculum Comparison

Current		Proposed	
Programming:	3 courses	Systems core:	3 courses
Theory: (2 + 2 depth):	4 courses	Theory core:	3 courses
Systems depth:	3 courses	Track (depth):	4-5 courses
Applications:	2 courses		
Electives:	2 courses	Electives:	2-4 courses
Capstone:	1 course	Capstone:	1 course
TOTAL:	15 courses	TOTAL:	14-15 courses

Same number of units in both cases

Outline

- The Computer Science student pipeline
- Misperceptions of the field
- CS curriculum revision at Stanford
  - Structure
  - Details
  - Issues
- Future considerations

Issues We've Considered

- *Undergraduate curriculum isn't broken, why fix it?*
  - Just because it's not broken, doesn't mean it can't be improved
  - Important to always consider innovation in curriculum
- *Students won't know what to take*
  - They don't need to know right away (common core)
  - Can provide effective advising from faculty and staff
- *Students won't have sufficient exposure to choose track*
  - Each track has "gateway" course to allow student to sample area
  - Gateway courses all also count as electives (by design)
  - After sampling 4-5 gateway courses, if a student still can't decide, he/she can still graduate under *unspecialized* track

Additional Issues We've Considered

- *What if very few students take a particular track?*
  - Some MS concentrations are small, but still available
  - Provides useful feedback for future revisions
- *Students will take fewer systems courses and potentially have weaker programming skills*
  - There is still a substantial amount of programming in curriculum
    - Students will still be well prepared for a broad array of work
  - Not all students want to take high-intensity programming jobs
    - For students who do, the Systems track is likely to be popular
    - Note: Currently, 47% of MS students take Systems specialization

Addressing In-Flight Students

- Students can graduate under any curricular requirements since they entered Stanford
  - You can choose to graduate under old or new CS requirements
- *What if a student has already taken introductory courses?*
  - Have a set of "course equivalences" to allow graduation under new requirements, if one so chooses
- *What if a students has already taken most upper-division courses?*
  - Can choose to just complete the old requirements
  - If not too many upper-division courses already taken, then new curriculum may still be an attractive option

Course Transition Plan in 2008-09

Fall	Winter	Spring
CS106X (Systems I)	CS106B (Systems I)	CS106B/X (Systems I)
CS107 (last old version)		CS107 (new Systems II)
		CS110 (Systems III)
CS103A	CS103B	
	CS103 (Theory I)	CS109 (Theory II)
CS161 (current)	CS161 (current)	

- Existing CS103A/B sequence offered one last time
- Students having taken old CS107 can take CS110
- Content of CS161 will transition (slightly) in 2009-10

Course Equivalences

- CS106B/X — Same course
- CS107 — Current CS107 satisfies new CS107
- CS110 — If you've taken CS108 and (CS140 or CS143) prior to Spring 08-09, CS108 will satisfy CS110 (but you'll need 1 extra unit in track + electives)
  If you've taken CS108, but not (CS140 or CS143) prior to Spring 08-09, can count CS108 as elective, but still need to satisfy CS110 requirement
- CS103 — CS103A/B or CS103X satisfies new CS103. If you took CS103X you'll need 1 additional unit in track + electives
- CS109 — Stat116 (or equivalent) taken prior to Spring 2008-09 satisfies CS109
- CS161 — Current CS161 satisfies new CS161 (CS161 may eventually become 5 unit course)

Additional Ideas Under Consideration

- Speaker series (seminar) on “Great themes in CS”
- Primarily targeted at Freshman
- Faculty and industry speakers showcase the field
  - Robotics, Graphics, Biocomputation, Search, Social Networking, etc.




Google

facebook

- Venue for role model women and minority speakers
- Also discuss issues (career paths, offshoring, etc.)

Discussion: thoughts on program and other potential issues?