Computer Science Curriculum 2013

Original slides for this talk developed by Mehran Sahami

Outline

- The CS2013 Effort
  - Steering Committee
  - Charter and Themes
- CS2013 Final Report
  - Volume contents
  - The Body of Knowledge
  - Curricular Organization
  - Course Exemplars
  - Curricular Exemplars
- Dissemination

CS2013 Steering Committee

ACM

- Mehran Sahami, Chair (Stanford)
- Andrea Danyluk (Williams College)
- Sally Fincher (Univ. of Kent)
- Kathleen Fisher (Tufts University)
- Dan Grossman (Univ. of Washington)
- Beth Hawthorne (Union County Coll.)
- Randy Katz (UC Berkeley)
- Rich LeBlanc (Seattle Univ.)
- Dave Reed (Creighton Univ.)

IEEE-CS

- Steve Roach, Chair (Exelis Inc.)
- Ernesto Cuadros-Vargas (Universidad Católica San Pablo, Peru)
- Ronald Dodge (US Military Academy)
- Robert France (Colorado State Univ.)
- Amruth Kumar (Ramapo College of NJ)
- Brian Robinson (ABB corporation)
- Remzi Seker (Embry-Riddle Aero. Univ.)
- Alfred Thompson (Microsoft, *retired*)

CS2013 Charter

*To review the Joint ACM and IEEE/CS Computer Science volume of Computing Curricula 2001 and the accompanying interim review CS 2008, and develop a revised and enhanced version for the year 2013 that will **match the latest developments in the discipline and have lasting impact**.*

*The CS2013 task force will seek **input from a diverse audience** with the goal of **broadening participation** in computer science. The report will seek to be **international in scope** and offer curricular and pedagogical guidance applicable to a **wide range of institutions**. The process of producing the final report will include multiple opportunities for public consultation and scrutiny.*

Timeline

- Fall 2010: Steering committee formed
  - Survey sent to 3500 department chairs
  - Revision to Body of Knowledge begins
  - Input solicited on Characteristics of CS Graduates
- Feb 2012: Strawman draft (alpha) public release
  - Includes: Body of Knowledge, Characteristics of Graduates
- Feb 2013: Ironman draft (beta) public release
  - Incorporates feedback received on Strawman draft
  - Includes: Complete draft of report
    - Additional course/curricular exemplars to be added later
- Dec 2013: Final report released
  - Fall 2013: ACM and IEEE-CS formally endorse final report

High-Level Themes of CS2013 Effort

- "Big Tent" view of Computer Science
  - "Outward" looking view of the field
  - Able to bridge to multi-disciplinary work ("Computational X")
- Managing curriculum size
  - Aim to not increase required hours from CC2001
  - Greater flexibility with respect to local needs/resources
- Course exemplars as opposed to stylized courses
  - Pointers to existing courses that incorporate knowledge units
  - Not creating a set of stylized reference classes
- Be aware of institutional needs
  - Variable goals, resources, and constraints
  - Variety of school sizes, school types, and available resources

CS2013 Contents

- Chapter 1: Introduction
- Chapter 2: Principles
- Chapter 3: Characteristics of Graduates
- Chapter 4: Introduction to the Body of Knowledge
- Chapter 5: Introductory Courses
- Chapter 6: Institutional Challenges
- Appendix A: The Body of Knowledge
- Appendix B: Migrating Curricula to CS2013
- Appendix C: Course Exemplars
- Appendix D Curricular Exemplars

514 pages total

Updating the Body of Knowledge

- Complete update of Body of Knowledge
  - Deemed most important in survey of department chairs
  - Drives discussion of pedagogy and complete curriculum
- Process for updating Body of Knowledge
  - Active subcommittee for each Knowledge Area
  - Chaired by a member of steering committee
    - Contains at least two other members of steering committee
    - Often contain additional (non-steering committee) members
  - Each area reviewed by several (often 4 or more) "external" reviewers prior to release of first (Strawman) draft
- Feedback from two preliminary drafts incorporated into final
  - Well over 100 (closer to 200) external reviewers involved

Knowledge Areas in CS2013

- AL - Algorithms and Complexity
- AR - Architecture and Organization
- CN - Computational Science
- DS - Discrete Structures
- GV - Graphics and Visual Computing
- HCI - Human-Computer Interaction
- **IAS - Information Assurance and Security**
- IM - Information Management
- IS - Intelligent Systems
- **NC - Networking and Communications**
- OS - Operating Systems
- **PBD - Platform-based Development**
- **PD - Parallel and Distributed Computing**
- PL - Programming Languages
- **SDF - Software Development Fundamentals**
- SE - Software Engineering
- **SF - System Fundamentals**
- SP - Social and Professional Issues

Body of Knowledge Update (Part 1)

- Two "foundational" KAs
  - Software Development Fundamentals
    - Includes content from old Programming Fundamentals, Software Engineering, and Algorithms and Complexity areas
    - Identifies foundational (paradigm-independent) concepts and skills (paradigms moved to Programming Languages)
    - Seeks to broaden thinking away from equating "Programming Fundamentals" with introductory programming courses (CS1,2)
  - Systems Fundamentals
    - Includes content from old Operating Systems, Architecture and Organization, and Algorithms and Complexity areas
    - Cross-cutting systems concepts (e.g., caching, locality, latency)
    - Avoids tying these to any one topic (e.g. Operating Systems, Architecture) to foster broader thinking and new pedagogy

Body of Knowledge Update (Part 2)

- Other new Knowledge Areas
  - Information Assurance and Security
    - Most important area to add based on survey of dept. chairs
    - Includes additional core curricular hours
  - Parallel and Distributed Computing
    - Second most important area to add based on survey of chairs
    - Includes additional core curricular hours
  - Networking and Communications (replaces Net-Centric Computing)
    - Sharpens focus on networking
    - Web development moves to "Platform-based Development"
  - Platform-based Development (elective only)
    - E.g., web, mobile devices, game consoles, robots, etc.

Curricular Organization (Part 1)

- Three-tiered classification of Body of Knowledge topics
  - **Core-Tier1**: essential topics, all of which are required for an undergraduate CS program
  - **Core-Tier2**: important foundational topics, the vast majority (at least 80%) of which should be in a student's CS program
    - Still considered "Core" topics – ideally all Tier2 topics would be included in an undergraduate program, if possible
    - Tiering allows for flexibility to locally customize curricula
  - **Elective**: additional topics that can be included to complete an undergraduate CS program
    - Covering just "core" material is insufficient for a complete curriculum

Curricular Organization (Part 2)

- Guidance provided on depth of coverage for learning outcomes in each Knowledge Area
  - 3 levels of depth: *Familiarity*, *Usage*, and *Assessment*
  - Familiarity: know what it means
  - Usage: can apply concept (e.g., write the code to use it)
  - Assessment: can compare/contrast/select appropriate method/ strategy for different situations
- Knowledge Areas are ***not*** necessarily courses
  - For example, introductory programming course might include: Software Development Fundamentals (key concepts) + Programming Languages (paradigm/language) + Platform (e.g., mobile devices or robots)

Example of Knowledge Area

Parallel and Distributed Computing (PD)

The past decade has brought explosive growth in multiprocessor computing, including multi-core processors and distributed data centers. As a result, parallel and distributed computing has moved from a largely elective topic to become more of a core component…

PD. Parallel and Distributed Computing (5 Core-Tier1 hours, 10 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
PD/Parallelism Fundamentals	2		N
PD/Parallel Decomposition	1	3	N
PD/Communication and Coordination	1	3	Y
PD/Parallel Algorithms, Analysis, and Programming		3	Y
PD/Parallel Architecture	1	1	Y
PD/Parallel Performance			Y
PD/Distributed Systems			Y
PD/Cloud Computing			Y
PD/Formal Models and Semantics			Y

Example of Knowledge Unit (Topics)

[Core-Tier1]

- Shared memory
- Consistency, and its role in programming language guarantees for data-race free programs

[Core-Tier2]

- Message passing
  - …
- Atomicity

…

[Elective]

- Consensus

Example KU Learning Outcomes

[Core-Tier1]

1. Use mutual exclusion to avoid a given race condition [Usage]
3. Give an example of an ordering of accesses among concurrent activities that is not sequentially consistent [Familiarity]

[Core-Tier2]

3. Give an example of a scenario in which blocking message sends can deadlock. [Usage]
4. Explain when and why multicast or event-based messaging can be preferable to alternatives [Familiarity]

…

[Elective]

12. Use semaphores or condition variables to block threads until a necessary precondition holds [Usage]

Bounding Size of Curriculum

	CS2013	
Knowledge Area	Tier1	Tier2
AL-Algorithms and Complexity	19	9
AR-Architecture and Organization	0	16
CN-Computational Science	1	0
DS-Discrete Structures	37	4
GV-Graphics and Visual Computing	2	1
HC-Human-Computer Interaction	4	4
IAS-Security and Information Assurance	3	6
IM-Information Management	1	9
IS-Intelligent Systems	0	10
NC-Networking and Communication	3	7
OS-Operating Systems	4	11
PBD-Platform-based Development	0	0
PD-Parallel and Distributed Computing	5	10
PL-Programming Languages	8	20
SDF-Software Development Fundamentals	43	0
SE-Software Engineering	6	22
SF-Systems Fundamentals	18	9
SP-Social and Professional Issues	11	5
Total Core Hours	**165**	**143**

All Tier1 + All Tier2 Total	**308**
All Tier1 + 90% of Tier2 Total	**293.7**
All Tier1 + 80% of Tier2 Total	**279.4**

Bounding Size of Curriculum

	CS2013		CS2008	CC2001	2007
Knowledge Area	Tier1	Tier2	Core	Core	LACS
AL-Algorithms and Complexity	19	9	31	31	69
AR-Architecture and Organization	0	16	36	36	40
CN-Computational Science	1	0	0	0	0
DS-Discrete Structures	37	4	43	43	49
GV-Graphics and Visual Computing	2	1	3	3	0
HC-Human-Computer Interaction	4	4	8	8	5
IAS-Security and Information Assurance	3	6	0	0	0
IM-Information Management	1	9	11	10	0
IS-Intelligent Systems	0	10	10	10	4
NC-Networking and Communication	3	7	15	15	10
OS-Operating Systems	4	11	18	18	9
PBD-Platform-based Development	0	0	0	0	0
PD-Parallel and Distributed Computing	5	10	0	0	0
PL-Programming Languages	8	20	21	21	47
SDF-Software Development Fundamentals	43	0	47	38	39
SE-Software Engineering	6	22	31	31	20
SF-Systems Fundamentals	18	9	0	0	0
SP-Social and Professional Issues	11	5	16	16	11
Total Core Hours	**165**	**143**	**290**	**280**	**303**

All Tier1 + All Tier2 Total	**308**
All Tier1 + 90% of Tier2 Total	**293.7**
All Tier1 + 80% of Tier2 Total	**279.4**

Introductory Courses (Chapter 5)

- Introductory courses much more diverse than CC2001
  - Rather than identify a small set of approaches, considered *design dimensions* for such courses
- Design dimensions
  - Pathways through introductory courses
  - Programming focus
  - Programming paradigm and choice of language
  - Software development practices
  - Parallel processing
  - Platform
- Each design dimension has discussion of options and explicit tradeoffs in different choices

Institutional Challenges (Chapter 6)

- Discussion of institutional issues
- Also a "catch-all" for topics we wanted to advocate
  - Localizing CS2013
  - Actively promoting computer science
  - Broadening participation
  - Computer Science across campus
  - Computer Science minors
  - Mathematics requirements in Computer Science
  - Computing resources
  - Maintaining a flexible and healthy faculty
  - Teaching faculty
  - Undergraduate teaching assistants
  - Online education

Course and Curricular Exemplars

- Process
  - Courses and curricula recommended or sought out
  - Some volunteered submissions
- Diversity and context
  - Variety of institutions represented
- These are examples to spur thinking in curriculum design
  - They are not meant to be the "best" (nor only) way
- CS2013 contains over 80 course exemplars from institutions around the world
- Also contains 5 full curricular exemplars
  - Bluegrass Community College (A.S. and A.A.S. degrees)
  - Grinnell College and Williams College
  - Stanford University

Course Exemplar Template

Name of Course, Location of Institution, Instructor, Course URL

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	Total Hours of Coverage
Name (e.g., Systems Fundamentals (SF))	*Number*

Where does the course fit in your curriculum?

What is covered in the course? (Short description, and/or a concise list of topics)

What is the format of the course? (Contact hours? Lecture, lab, discussion?)

How are students assessed? (Assignments: number and type, expected workload)

Course textbooks and materials

Why do you teach the course this way? (Course rationale and goals)

Body of Knowledge coverage

KA	Knowledge Unit	Topics Covered	Hours
XY	*Full name of KU*		*Num*

Additional topics/comments

Curricular Exemplar Template

Name of School, Name of Department, Department URL, Contact Person

Curricular Overview and Analysis

Computer Science Major

Percentage of Core-Tier1 and Core-Tier2 Topics Covered in Program

Knowledge Units in Typical Major

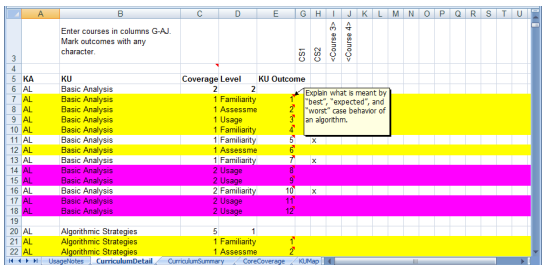
		Course 1	Course 2	Course 3	Course 4	% Tier 1	% Tier 2
AL	Basic Analysis						
	Algorithmic Strategies						
	Fund. Data Struct. & Algorithms						
	Basic Autom.ata & Complexity						
AR	Digital Logic						
	Machine-level rep. of data						
	Assembly level machine organiz.						
	Memory org. and arch.						
	Interfacing and communication						

Planned Curricular Revisions (Optional)

Information in Individual Courses

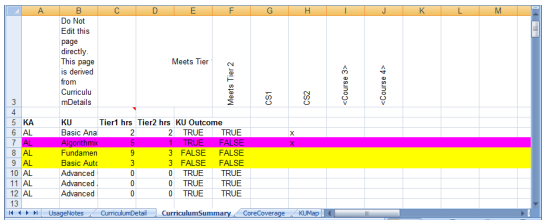
Mapping Your Curriculum (Part I)

- Curricular mapping spreadsheet on cs2013.org website
  - Organized by Knowledge Area and Unit
  - Curriculum Detail lists *all* learning outcomes in each area



Mapping Your Curriculum (Part II)

- Curricular Summary sheet in spreadsheet
  - Shows summary of Knowledge Areas and Units
  - Color coded based on Details sheet to indicate if Knowledge Units that satisfy Core-Tier1 and Core-Tier2 outcomes



	A	B	C	D	E	F	G	H	I	J	K	L	M
3		Do Not Edit this page directly. This page is derived from CurriculumDetails			Meets Tier	Meets Tier 2	CS1	CS2	<Course 3>	<Course 4>			
4													
5	KA	KU	Tier1 hrs	Tier2 hrs	KU Outcome								
6	AL	Basic Ana	2	2	TRUE	TRUE		x					
7	AL	Algorithmi	5	1	TRUE	FALSE		x					
8	AL	Fundamen	9	3	FALSE	FALSE							
9	AL	Basic Aut	3	3	FALSE	FALSE							
10	AL	Advanced	0	0	TRUE	TRUE							
11	AL	Advanced	0	0	TRUE	TRUE							
12	AL	Advanced	0	0	TRUE	TRUE							
13													

UsageNotes | CurriculumDetail | CurriculumSummary | CoreCoverage | KUMap

Final Report and Mapping Spreadsheets available at

www.cs2013.org