Computer Science Curriculum 2013: Curricular Guidelines for the Next Decade

Mehran Sahami  
Stanford University  
sahami@cs.stanford.edu

Dan Grossman  
University of Washington  
djg@cs.washington.edu

Rich LeBlanc  
Seattle University  
leblanc@seattleu.edu

Remzi Seker  
Univ. of Arkansas, Little Rock  
rxseker@ualr.edu
Outline

• Computing Curriculum background
• The CS2013 Effort
  – Steering Committee
  – Charter and Themes
• CS2013 Strawman Report
  – Volume Contents
  – Characteristics of CS Graduates
  – The Body of Knowledge
  – Curricular Organization
• Community Engagement
  – Discussion
Computing Curriculum Background

• Every decade, ACM and IEEE-Computer Society jointly sponsor a curricular volume on Computer Science
  – Aimed at providing modern curricular guidance for undergraduate Computer Science programs internationally
  – Starting in 2001, volumes splits by disciplines:
    – Computer Science (CS), Computer Engineering (CE), Information Systems (IS), Information Technology (IT), and Software Engineering (SE)
    – Modest “interim” revision of CS volume was released in 2008

• Next full CS volume is set for release in 2013
  – Hence the name “CS2013”
  – Work on this volume began in Fall 2010
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)

**IEEE-CS**
- Steve Roach, Chair (U. of Texas, El Paso)
- Ernesto Cuadros-Vargas (Universidad Católica San Pablo, Peru)
- Ronald Dodge (US Military Academy)
- Robert France (Colorado State)
- Amruth Kumar (Ramapo College of NJ)
- Brian Robinson (ABB corporation)
- Remzi Seker (U. of Arkansas, Little Rock)
- Alfred Thompson (Microsoft)
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.
High-Level Themes of CS2013 Effort

- “Big Tent” view of Computer Science
  - “Outward” looking view of the field
  - Making room for 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
Principles for CS2013

1. Identify **essential skills** and **body of knowledge** for CS undergraduates.
2. CS is rapidly changing field, drawing from and contributing to variety of disciplines. Prepare students for **lifelong learning**.
3. CS2013 is serving **many constituents**, including: faculty, students, administrators, curricula developers, and industry.
4. Curricular guidelines must be relevant to a **variety of institution types** (large/small, research/teaching, 4-yr/2-yr, US/int’l)
5. Provide **guidance for level of mastery for topics**, and show exemplars of fielded courses covering topics.
6. Provide **realistic, adoptable recommendations** that support novel curricular designs, and attract full range of talent to field.
7. Should include **professional practice** (e.g. communication skills, teamwork, ethics) as part of undergraduate experience.
CS2013 Contents: Strawman Report

• Guiding principles
• Body of knowledge
  – Topically organized set of “Knowledge Areas”
  – Knowledge Areas provide list of topics and learning outcomes
• Curricular structure
  – Guidance on how Body of Knowledge translates into curriculum
• Professional considerations
  – Characteristics of CS graduates
CS2013 Contents: Plan for Final Report

• Guiding principles
• Body of knowledge
  – Topically organized set of “Knowledge Areas”
  – Knowledge Areas provide list of topics and learning outcomes
• Curricular structure
  – Guidance on how Body of Knowledge translates into curriculum
  – Institutional challenges
• Professional considerations
  – Characteristics of CS graduates
  – Professional practice
• Course and curricular exemplars
  – Pointers to and discussion of example curricula/courses reflecting diverse ways of covering the Body of Knowledge
Characteristics of CS Graduates

- Technical understanding of Computer Science
- Familiarity with common themes and principles
- Appreciation of the interplay between theory and practice
- System-level perspective
- Problem solving skills
- Project experience
- Commitment to life-long learning
- Commitment to professional responsibility
- Communication and organizational skills
- Awareness of the broad applicability of computing
- Appreciation of domain-specific knowledge
Professional Practice

• Computing professionals need training beyond technical skills, including:
  – Ethical reasoning and legal responsibility
  – Communication (written and oral)
  – Teamwork
  – Project management

• Chapter on Professional Practice will be forthcoming in future draft
  – Welcome thoughts to help develop this area
Updating the Body of Knowledge

• Strawman report: 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 Strawman draft
  – Over 100 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
- HC - 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 Units
  - **Core-Tier1**: absolutely essential topics, all of which are required for any undergraduate CS program
  - **Core-Tier2**: important foundational topics, the vast majority (no less than 80%) of which should be in a 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: *Knowledge*, *Application*, and *Evaluation*
  – *Knowledge*: know what it means
  – *Application*: can apply concept (e.g., write the code to use it)
  – *Evaluation*: 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, 9 Core-Tier2 hours)

<table>
<thead>
<tr>
<th>Course</th>
<th>Core-Tier1 hours</th>
<th>Core-Tier2 hours</th>
<th>Includes Electives</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD/Parallelism Fundamentals</td>
<td>2</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>PD/Parallel Decomposition</td>
<td>1</td>
<td>3</td>
<td>N</td>
</tr>
<tr>
<td>PD/Communication and Coordination</td>
<td>1</td>
<td>3</td>
<td>Y</td>
</tr>
<tr>
<td>PD/Parallel Algorithms, Analysis, and Programming</td>
<td></td>
<td>3</td>
<td>Y</td>
</tr>
<tr>
<td>PD/Parallel Architecture</td>
<td>1</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>PD/Parallel Performance</td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>PD/Distributed Systems</td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>PD/Formal Models and Semantics</td>
<td></td>
<td></td>
<td>Y</td>
</tr>
</tbody>
</table>
Example of Knowledge Unit (Topics)

PD/Communication and Coordination

[1 Core-Tier1 hour, 3 Core-Tier2 hours]

Topics:

[Core-Tier1]
- Shared Memory
- Sequential consistency, and its role in programming language guarantees for data-race-free programs

[Core-Tier2]
- Consistency in shared memory models
- Message passing
- Point-to-point versus multicast (or event-based) messages
- Blocking versus non-blocking styles for sending and receiving messages
- Message buffering (cross-reference PF/Fundamental Data Structures/Queues)
- Atomicity
  ...

[Elective]
- Consensus
- (Cyclic) barriers, counters, or related constructs
  ...
Example KU Learning Outcomes

1. Use mutual exclusion to avoid a given race condition [Application]
2. Give an example of an ordering of accesses among concurrent activities that is not sequentially consistent [Knowledge]
3. Explain when and why multicast or event-based messaging can be preferable to alternatives [Knowledge]
4. Write a program that correctly terminates when all of a set of concurrent tasks have completed [Application]
5. Use a properly synchronized queue to buffer data passed among activities [Application]

...
# Bounding Size of Curriculum

<table>
<thead>
<tr>
<th>Knowledge Area</th>
<th>CS2013 Tier1</th>
<th>Tier2</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL-Algorithms and Complexity</td>
<td>19</td>
<td>9</td>
</tr>
<tr>
<td>AR-Architecture and Organization</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>CN-Computational Science</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>DS-Discrete Structures</td>
<td>37</td>
<td>4</td>
</tr>
<tr>
<td>GV-Graphics and Visual Computing</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>HC-Human-Computer Interaction</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>IAS-Security and Information Assurance</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>IM-Information Management</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>IS-Intelligent Systems</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>NC-Networking and Communication</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>OS-Operating Systems</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>PBD-Platform-based Development</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PD-Parallel and Distributed Computing</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>PL-Programming Languages</td>
<td>8</td>
<td>20</td>
</tr>
<tr>
<td>SDF-Software Development Fundamentals</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>SE-Software Engineering</td>
<td>6</td>
<td>21</td>
</tr>
<tr>
<td>SF-Systems Fundamentals</td>
<td>18</td>
<td>9</td>
</tr>
<tr>
<td>SP-Social and Professional Issues</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td><strong>Total Core Hours</strong></td>
<td><strong>163</strong></td>
<td><strong>142</strong></td>
</tr>
</tbody>
</table>

| All Tier1 + All Tier2 Total          | 305          |
| All Tier1 + 90% of Tier2 Total       | 290.8        |
| All Tier1 + 80% of Tier2 Total       | 276.6        |
## Bounding Size of Curriculum

<table>
<thead>
<tr>
<th>Knowledge Area</th>
<th>CS2013 Tier1</th>
<th>Tier2</th>
<th>CS2008 Core</th>
<th>CC2001 Core</th>
<th>2007 LACS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL-Algorithms and Complexity</td>
<td>19</td>
<td>9</td>
<td>31</td>
<td>31</td>
<td>69</td>
</tr>
<tr>
<td>AR-Architecture and Organization</td>
<td>0</td>
<td>16</td>
<td>36</td>
<td>36</td>
<td>40</td>
</tr>
<tr>
<td>CN-Computational Science</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DS-Discrete Structures</td>
<td>37</td>
<td>4</td>
<td>43</td>
<td>43</td>
<td>49</td>
</tr>
<tr>
<td>GV-Graphics and Visual Computing</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>HC-Human-Computer Interaction</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>IAS-Security and Information Assurance</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IM-Information Management</td>
<td>1</td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>IS-Intelligent Systems</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>NC-Networking and Communication</td>
<td>3</td>
<td>7</td>
<td>15</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>OS-Operating Systems</td>
<td>4</td>
<td>11</td>
<td>18</td>
<td>18</td>
<td>9</td>
</tr>
<tr>
<td>PBD-Platform-based Development</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PD-Parallel and Distributed Computing</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PL-Programming Languages</td>
<td>8</td>
<td>20</td>
<td>21</td>
<td>21</td>
<td>47</td>
</tr>
<tr>
<td>SDF-Software Development Fundamentals</td>
<td>42</td>
<td>0</td>
<td>47</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>SE-Software Engineering</td>
<td>6</td>
<td>21</td>
<td>31</td>
<td>31</td>
<td>20</td>
</tr>
<tr>
<td>SF-Systems Fundamentals</td>
<td>18</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SP-Social and Professional Issues</td>
<td>11</td>
<td>5</td>
<td>16</td>
<td>16</td>
<td>11</td>
</tr>
<tr>
<td><strong>Total Core Hours</strong></td>
<td><strong>163</strong></td>
<td><strong>142</strong></td>
<td><strong>290</strong></td>
<td><strong>280</strong></td>
<td><strong>303</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>305</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Tier1 + 90% of Tier2 Total</td>
<td>290.8</td>
</tr>
<tr>
<td>All Tier1 + 80% of Tier2 Total</td>
<td>276.6</td>
</tr>
</tbody>
</table>
Engaging the Community

- Website: cs2013.org
  - Dissemination of report drafts (Strawman report available)
  - Community engagement
    - Use Ensemble for commenting/feedback
- Multiple opportunities for involvement in this effort
  - Comments on Strawman draft
  - Mapping exemplar courses/curricula to Body of Knowledge
  - Pedagogic approaches and instructional designs
    - Address professional practice within undergraduate curricula
  - Share institutional challenges (and solutions to them)
  - Suggest roles that can contribute to this effort
Timeline

• Feb. 2012: Strawman draft (alpha) public release
  – Includes: Body of Knowledge, Characteristics of Graduates

• July 2012: Comment period for Strawman closes
  – Next steering committee meeting to discuss feedback starts August 2nd

• Feb. 2013: Ironman draft (beta) public release
  – Incorporates feedback on Strawman draft
  – Includes: Body of Knowledge, Characteristics of Graduates, Curricula and Course Exemplars, Professional Practice, Institutional Challenges

• June 2013: Comment period for Ironman draft closes

• Summer 2013: Final report released (general availability)
Feedback on CS2013

• Comments on the Strawman report?
  – Positive and negative reactions
  – Issues of Characteristic of CS Graduate/Professional Practice
    • Especially the role of project experiences
  – Thoughts on the Body of Knowledge

• What might help with adopting new curricular recommendations in CS2013?
  – What are institutional challenges (and potential solutions)?

• Additional comments?