Computing Curriculum 2001

Richard H. Eckhouse
Math & Computer Science
UMass Boston
eckhouse@cs.umb.edu
Charter: To review the Joint ACM and IEEE/CS Computing Curricula 1991 and develop a revised and enhanced version for the year 2001 that will match the latest developments of computing technologies in the past decade and endure through the next decade.

Target date for final draft: summer 2001

http://www.computer.org/education/cc2001
### ACM

*Education Board chair:*
- Peter Denning

*Task Force co-chairs:*
- Eric Roberts (*editor*)
- Russ Shackelford

*Steering committee members:*
- Richard Austing
- Fay Cover
- Gordon Davies
- Andrew McGettrick
- Michael Schneider
- Ursula Wolz

### IEEE Computer Society

*VP for Education:*
- Carl Chang

*Task Force co-chairs:*
- James Cross
- Gerald Engel (*editor*)

*Steering committee members:*
- Doris Carver
- Dick Eckhouse
- Willis King
- Francis Lau
- Robert Sloan (*secretary*)
- Pradip Srimani
History of Curriculum Reports

- 1967 - *COSINE report (Commission on Engineering Education)*
- 1968 - *Curriculum ’68 (ACM)*
- 1977 - *A Curriculum in CS and Engineering (IEEE-CS)*
- 1978 - *Curriculum ’78 (ACM)*
- 1983 - *Model Program in CS and Engineering (IEEE-CS)*
- 1989 - *Computing as a Discipline*
- 1991 - *Computing Curricula ’91 (IEEE-CS + ACM)*
Computing Curricula 2001
Meeting Schedule

- Initial meeting (Chicago, November 1998)
- Coordinating session with chairs (Atlanta, April 1999)
- Steering Committee (Richmond, June 1999)
- Steering Committee (Palo Alto, August 1999)
- Coordinating session (San Juan, November 1999)
- Steering Committee (Colorado Springs, January 2000)
- Coordinating session (Boston, February 2000)
- Coordinating session (Austin, March 2000)
- Steering Committee (Washington, April 2000)
- Steering Committee + focus groups (Minneapolis, June 2000)
- Steering Committee (Los Alamitos, August 2000)
- Steering Committee (Kansas City, October 2000)
- Steering Committee (Palo Alto, December 2000)
Computing Curricula 2001
Accomplishments

- Completed a survey and evaluation of the impact of CC’91
- Assessed the major changes in the discipline
- Articulated a set of principles to guide our work
- Developed an organizational structure and strategy
- Established knowledge area focus groups
- Created pedagogy focus groups to consider broad issues
- Reviewed the reports of those working groups
- Drafted a body of knowledge for computer science
- Proposed a set of core topics for undergraduates in CS
- Conducted a working session for the PFGs and KFG chairs
- Compiled a preliminary set of course syllabi
- Created two working drafts (Straw Man and Iron Man)
Computing Curricula 2001
Pedagogy Focus Groups

- Introductory topics and courses
- Supporting topics and courses
- The computing core
- Professional practices
- Advanced study and undergraduate research
- Computing across curricula
Computing Curricula 2001

Principles

1. Computing has become an extremely broad discipline that extends well beyond the traditional boundaries of computer science.
2. Despite its growing breadth, computing remains an integrated field of study that draws its foundations from many well-established disciplines.
3. The rapid evolution of the computing discipline requires an ongoing review of the corresponding curriculum.
4. Curriculum 2001 must go beyond knowledge units to offer significant guidance in terms of individual course design.
Computing Curricula 2001
Principles (Cont’d)

5. Curriculum 2001 must identify a relatively small set of core concepts and skills that are required of all computing students.
6. Curriculum 2001 must provide guidelines for courses beyond the required core.
7. Curriculum 2001 must be international in scope.
8. The development of Curriculum 2001 must involve significant industry participation.
9. Curriculum 2001 must include professional practice as an integral component of the undergraduate curriculum.
10. Curriculum 2001 must be useful for its intended audience.
The overview document is common to all computing disciplines and describes the general principles that underlie the specific disciplinary reports.

The reports on the Computer Science and Computer Engineering will be published by the CC2001 Task Force itself as part of its overall charter.

These reports—perhaps with additional volumes for other disciplines—will be prepared in consultation with existing curriculum committees in these areas. In many cases, these committees have already published curriculum guidelines that can easily be incorporated into the CC2001 structure.
Outline for Overview

Volume I

1. Introduction
   1.1 Computing and change
      1.1.1 Technological changes
      1.1.2 Cultural changes
   1.2 The expansion of the discipline
      1.2.1 Overview of the major subfields
      1.2.2 Commonalities among the fields
   1.3 Defining a curriculum
      1.3.1 Defining the body of knowledge
      1.3.2 Defining the pedagogical framework
      1.3.3 Strategy and tactics

Bibliography
<table>
<thead>
<tr>
<th>Sections</th>
<th>Volume II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive summary</td>
<td></td>
</tr>
<tr>
<td>1. Introduction</td>
<td></td>
</tr>
<tr>
<td>2. Lessons from past reports</td>
<td></td>
</tr>
<tr>
<td>3. Changes in the CS Body of Knowledge</td>
<td></td>
</tr>
<tr>
<td>4. Principles</td>
<td></td>
</tr>
<tr>
<td>5. Overview of the CS Body of Knowledge</td>
<td></td>
</tr>
<tr>
<td>6. Overview of the Curricular Models</td>
<td></td>
</tr>
<tr>
<td>7. Introductory Courses</td>
<td>9. Advanced Courses</td>
</tr>
<tr>
<td>8. Intermediate Courses</td>
<td>10. Professional Practice</td>
</tr>
<tr>
<td></td>
<td>11. Characteristics of CS graduates</td>
</tr>
<tr>
<td></td>
<td>12. General Requirements</td>
</tr>
<tr>
<td></td>
<td>13. Computing across the Curriculum</td>
</tr>
<tr>
<td></td>
<td>14. Strategy and Tactics</td>
</tr>
<tr>
<td></td>
<td>Acknowledgments</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
</tr>
<tr>
<td></td>
<td>A. The CS Body of Knowledge</td>
</tr>
<tr>
<td></td>
<td>B. CS course descriptions</td>
</tr>
</tbody>
</table>

January 2001
## The Undergraduate CS Core

- DS. Discrete Structures 43 core hours
- PF. Programming Fundamentals 54 core hours
- AL. Algorithms and Complexity 31 core hours
- PL. Programming Languages 6 core hours
- AR. Architecture and Organization 36 core hours
- OS. Operating Systems 18 core hours
- HC. Human-Computer Interaction 6 core hours
- GV. Graphics and Visualization 5 core hours
- IS. Intelligent Systems 10 core hours
- IM. Information Management 10 core hours
- NC. Net-Centric Computing 15 core hours
- SE. Software Engineering 30 core hours
- SP. Social and Professional Issues 16 core hours
- Total 280 core hours
Changes in the CS Core from CC’91

- The required core is now smaller (280 vs. ~323 hours)
- Coverage of architecture, algorithms, operating systems, and programming languages has been reduced
- Coverage of graphics, HCI, and networking has been expanded
- We have emphasized that the core no longer constitutes a complete curriculum, but must be supplemented by additional topics appropriate to the specific program
## Structure of the Curricular Options

<table>
<thead>
<tr>
<th>Introductory courses</th>
<th>Imperative first</th>
<th>Objects first</th>
<th>Functional first</th>
<th>Breadth first</th>
<th>Algorithms first</th>
<th>Hardware first</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transition</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific material needed to match introductory and intermediate course strategies (possibly an additional course and minor adjustments to topic coverage)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Intermediate courses</strong></td>
<td>Topic-based approach</td>
<td>Compressed approach</td>
<td>Systems-based approach</td>
<td>Web-based approach</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Advanced courses</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Additional courses used to complete the undergraduate program</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Course Numbering Scheme

Level
1xx = introductory, 2xx = intermediate, 3xx = advanced

Subject area
0 = Programming fundamentals (PF)
1 = Theory/Languages (DS, AL, PL)
2 = Architecture (AR)
3 = Op. systems/Networks (OS, NC)
4 = AI/Info Management (IS, IM)
5 = HCI/Graphics (HC, GV)
6 = Professional practice (SE, SP)
7 = Applications (CN, other)
8 = Cross-cutting courses
9 = Project/independent work

Pedagogical approach (optional)
I = Imperative-first
O = Objects-first
F = Functional-first
B = Breadth-first
A = Algorithms-first
H = Hardware-first
T = Traditional discrete-topic-based approach
C = Compressed approach
S = Systems-based approach
W = Web-based approach

Identifying number within area
Sample Course Description

Course number . Course Title
Course description written in the style of a university course catalog

Prequisite:
Required courses, units, or background

Syllabus:
Bulleted list providing an outline of the topics covered

Units covered:
List of units from the CS Body of Knowledge

Notes:
Optional narrative section offering additional notes about the course
Iron Man draft to be released in February 2001
  - It is still under discussion
There will be a new web format that encourages feedback and discussion
  - That feedback will be actively managed so as to be responsive

QUESTIONS?