The New CSTA K-12 Standards

2011 Hopper Equity Workshop

Dr. Chris Stephenson
Why New Standards
Knowledge for Today and Beyond

We consider it critical that students be able to read and write and understand the fundamentals of math, biology, chemistry and physics. To be a well-educated citizen in today’s computing-intensive world, students must have a deeper understanding of the fundamentals of computing as well.
Very Scary Numbers

Schools Offering Introductory Computer Science

<table>
<thead>
<tr>
<th></th>
<th>2005</th>
<th>2007</th>
<th>2009</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td>78%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td>73%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>65%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011</td>
<td>69%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Schools Offering Advanced Placement Computer Science

<table>
<thead>
<tr>
<th></th>
<th>2005</th>
<th>2007</th>
<th>2009</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td>40%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td>32%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>27%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011</td>
<td>36%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Data from the CSTA National High School Computer Science Surveys.
Jobs and Graduates

Annual STEM Job Openings vs College Graduates Through 2018

The New K-12 Computer Science Learning Standards
Why Standards?

To be considered a valid academic discipline in K-12 computer science must have:

• Assessable descriptions of what the students should know and be able to do at each level of their educational experience
• A reasonable scope and sequence that matches expectations to intellectual/conceptual development
• An organizing structure that identifies conceptual themes relevant to the discipline and to all students
• A description of possible courses linked to student interests and educational destinations
• A standards document that looks like everyone else’s standards document
The ACM/CSTA Model Curriculum

• Published in:
  – 2003
  – 2006: revised forward
• More than 40,000 copies distributed
• Used as the basis for curriculum development at the state and national level
Model Curriculum Revised Committee

- Co-Chairs:
  - Deborah Seehorn, North Carolina Department of Public Instruction

- Committee Members:
  - Steve Carey, Brunswick School Department (ME)
  - Brian Fuschetto, Lyndhurst High School (NJ)
  - Irene Lee, Santa Fe Research Institute (NM)
  - Daniel Moix, College of the Oachitas (AR)
  - Dianne O'Grady-Cunniff, Howard High School (MD)
  - Barbara Boucher Owens: SIGCSE Board
  - Chris Stephenson, CSTA
  - Anita Verno, Bergen Community College (NJ)
Timelines

- April 1: Draft completed
- April 15: Community review and feedback process begins
- June 5: Revision process begins
- August 1: Second draft completed
- August 15: Community endorsement process begins
- September 16: Final draft to layout
- November 18: Final version to printer
- December 16: Publication
What We Learned from the First Review

• Lot of incredibly helpful suggestions (2/3 of suggested changes were made in second draft)
• Divergent perspectives of K-12 and university reviewers was immediately apparent
  — University people found it difficult to understand the strand organization
  — A few university people still object to “computational thinking”
• We needed to change a course name
• There were review perspective gaps that we needed to fill (K-8 specialists, assessment specialist, etc.)
• We added a second round of reviews by targeted K-12 educators who were willing to do very deep reviews
Organizing Structure

Level 1
Grades K-6
- Computer Science and Me

Level 2
Grades 6-9
- Computer Science and Community

Level 3
Grades 9-12
- Computer Science in the Modern World
- Computer Science Concepts and Practices
- Topics in Computer Science

Applying Concepts and Creating Real-World Solutions
Level Definitions

- **Level 1 (recommended for grades K–6) Computer Science and Me:** elementary school students are introduced to foundational concepts in computer science by integrating basic skills in technology with simple ideas about computational thinking. The learning experiences created from these standards should be inspiring and engaging, helping students see computing as an important part of their world.

- **Level 2 (recommended for grades 6–9) Computer Science and Community:** middle school/junior high school students begin using computational thinking as a problem-solving tool. They begin to appreciate the ubiquity of computing and the ways in which computer science facilitates communication and collaboration. Students begin to experience computational thinking as a means of addressing community-relevant issues. The learning experiences created from these standards should be relevant to the students and should promote their perceptions of themselves as proactive and empowered problem solvers.

- **Level 3 (recommended for grades 9–12) Applying concepts and creating real-world solutions:** Level 3 is divided into three discrete courses, each of which focuses on different facets of computer science as a discipline. Throughout these courses, students can master more advanced computer science concepts and apply those concepts to develop virtual and real-world artifacts. The learning experiences created from these standards should focus on the exploration of real world problems and the application of computational thinking to the development of solutions.
Level 3 Course Descriptions

• **Level 3A: (recommended for grades 9 or 10) Computer Science in the Modern World:** This course should be available to all students. Its goal is to solidify students’ understanding of computer science principles and practices so that they can make informed choices and use appropriate computational tools and techniques in whatever career they decide to pursue. They should also appreciate the breadth of computing and its influence in almost every aspect of modern life. Finally, they should understand the social and ethical impact of various choices when they are using computing technology in their work and personal lives.

• **Level 3B: (recommended for grades 10 or 11) Computer Science Principles:** This course is a more in-depth study of computer science and its relation to other disciplines, and contains a significant amount of algorithmic problem solving and related activities. One way to realize this course is by following the new AP Computer Science Principles course (www.apcsp.org). Students should come out of this course with a clear understanding of the application of computational thinking to real-world problems. They should also have learned how to work collaboratively to solve a problem and use modern collaboration tools during that work.

• **Level 3C: (recommended for grades 11 or 12) Topics in Computer Science:** This is an elective course that provides depth of study in one particular area of computing. This may be, for example, an AP Computer Science A (AP, 2010) course, which offers depth of study in Java programming. Alternatively, this offering may be a projects-based course focusing on a single facet of computing or a course that leads to professional computing certification. Any Level 3 course will typically require at least the Level 2 course as a prerequisite.
Strands

- Computational Thinking
- Community, Global, and Ethical Impacts
- Collaboration
- Computers and Communication Devices
- Computing Practice & Programming
Example Strand for Level 1

Computers and Communication Devices

Grades K–3
The student will be able to:
1. Use standard input and output devices to successfully operate computers and related technologies.

Grades 3–6
The student will be able to:
1. Demonstrate an appropriate level of proficiency with keyboards and other input and output devices.
2. Recognize that computers model intelligent behavior (as found in robotics, speech and language recognition, and computer animation).
3. Identify factors that distinguish humans from machines.
4. Understand the pervasiveness of computers and computing in daily life (e.g., voice mail, downloading videos and audio files, microwave ovens, thermostats, wireless Internet, mobile computing devices, GPS systems, etc.).
5. Apply strategies for identifying simple hardware and software problems that may occur during use.
6. Identify that information is coming to the computer from many sources over a network.
Example Strand for Level 2

Computing Practice & Programming / Careers

The student will be able to:

1. Use a variety of multimedia tools and peripherals to support personal productivity and learning throughout the curriculum.
2. Select appropriate tools and technology resources to accomplish a variety of tasks and solve problems.
3. Design, develop, publish, and present products (e.g., webpages, mobile applications, animations) using technology resources that demonstrate and communicate curriculum concepts.
4. Implement problem solutions using a programming language, including: looping behavior, conditional statements, logic, expressions, variables, and functions.
5. Demonstrate an understanding of algorithms and their practical application.
6. Demonstrate good practices in personal information security using passwords, encryption, and secure transactions.
7. Demonstrate dispositions amenable to open-ended problem solving and programming (e.g., comfort with complexity, persistence, brainstorming, adaptability, patience, propensity to tinker, creativity, accepting challenge).
8. Identify interdisciplinary careers that are enhanced by computer science.
9. Collect and analyze data that is output from multiple runs of a computer program.
Example Strand for Level 3A Course

Computational Thinking
The student will be able to:
1. Describe how various types of data are stored in a computer system.
2. Analyze the representation and trade-offs among various forms of digital information.
3. Describe the relationship between binary and hexadecimal representations.
4. Explain how sequence, selection, iteration, and recursion are building blocks of algorithms.
5. Describe a software development process used to solve software problems (e.g., design, coding, testing, verification).
6. Use predefined functions and parameters, classes and methods to divide a complex problem into simpler parts.
7. Use modeling and simulation to represent and understand natural phenomena.
8. Describe the concept of parallel processing as a strategy to solve large problems.
9. Discuss the value of abstraction to manage problem complexity.
10. Compare techniques for analyzing massive data collections.
11. Describe how computation shares features with art and music by translating human intention into an artifact.
Maps of the CSTA K-12 Standards
Maps of the CSTA K-12 Standards
Maps of the CSTA K-12 Standards
How These Standards Will Be Distributed

- Hypertext pdf on the CSTA website
- Hardcopy available for advocacy work
- Hardcopy at CSTA professional development events
- Mailing to curriculum superintendents in selected state departments of education
THANK YOU!

Chris Stephenson  
Executive Director, CSTA  
Phone/Fax: 1-541-687-1840  
cstephenson@csta.acm.org

CSTA website: csta.acm.org