Model Curriculum for K-12 Computer Science:
Draft Report of the ACM K-12 Task Force Curriculum Committee*†

Allen Tucker (Chair) - Bowdoin College
Fadi Deek - New Jersey Inst. of Technology
Jill Jones - Carl Hayden High School
Dennis McCowan - Weston Public Schools
Chris Stephenson - University of Waterloo
Anita Verno - Bergen Community College

* See www.acm.org/k12 for the complete draft report.
† Send all correspondence to allen@bowdoin.edu
Why a K-12 Computer Science (CS) Curriculum?

- Ubiquity of computers in the world
- Rapid evolution of CS as a discipline
- Need for improved public understanding of CS
- Workforce shortages in CS at all levels
- CS offers complementary academic skills
- Current curriculum (ACM 1993) is outdated
This Curriculum Design Process

- Form committee (Spring 02)
- Survey community views (3/02 - 6/02)
- Draft a model curriculum (7/02 - 11/02)
- Solicit broad feedback (11/02 - 6/03)
- Revise the draft (6/03 - 9/03)
- Publish the revised draft (9/03 - 12/03)
We are soliciting feedback from many groups...

- ACM Education Board
- ACM SIGCSE members
- ISTE Special Interest Group for Computer Science (SIGCS)
- Directors of curriculum in school districts (ASCD)
- National Education Association (NEA)
- National Association of Secondary School Principals (NASSP)
- National School Board Association
- Academy of Information Technology/National Academy Foundation (AOIT/NAT)
Computer Science and IT

- **Computer science (CS)** = the study of computational processes (algorithms)
- **Information Technology (IT)** = the study of technology that facilitates information processing
- CS and IT are fundamentally different, though they share some common ground
- In K-12, CS and IT have developed sporadically (at best)
- A new K-12 model curriculum should complement both
  - Existing IT curricula
  - Existing CS curricula (especially AP)
Goals of a New K-12 CS Model Curriculum

- to introduce the concepts of CS, beginning at K-8
- to teach high school CS in a way that is both
  - accessible to all students and
  - worthy of math/science credit
- to provide CS electives that prepare students for either
  - the technology work force or
  - higher education
- to be implemented in a majority of states and districts
Computer Science is a Mature Field

- Colleges and universities produce about 40,000 BS/BAs and 900 PhDs per year
- Undergraduate CS curricula have been evolving since 1968 (e.g., “Computing Curricula 2001”)
- From 1990 to 2000, computer science grew more rapidly (from 500,000 to 1.2 million workers) than any other science or engineering workforce category
- Oddly, K-12 computer science has not reached maturity (or even adolescence!) in most K-12 schools
Design of a New Model Curriculum

Grade

K-8
Level I - Preparation for Computer Science

9 or 10
Level II - CS in the Modern World

10 or 11
Level III - CS as Analysis and Design

11 or 12
Level IV - Topics in Computer Science
Level I - Preparation for Computer Science (Grades K-8)

Purpose: To prepare students for computer science
- basic skills in technology use
- simple ideas about algorithmic thinking

Method:
- specific courses, or
- short modules within science, math or social studies courses
Level I Content

National Educational Technology Standards (NETS) +
Introduction to Algorithmic Thinking

Why NETS?:
- carefully defined by ISTE
- widely endorsed

Why More?
- algorithmic thinking not a traditional part of K-8
- focuses on problem solving and logical thinking
- introduces core computer science concepts
Level I Outcomes

Students should become

- computer users,
- information seekers, analyzers, and evaluators,
- aware of multi-step algorithmic solutions,
- problem solvers and decision makers,
- users of productivity tools,
- information producers and communicators
Level II - Computer Science in the Modern World (Grade 9 or 10)

Purpose: To provide an introduction to the principles of computer science and its place in the modern world. (this is a required course for all students)

Prerequisites:
- ability to use a computer and the Internet,
- understanding of Level 1 concepts (file creation, databases, spreadsheets, graphics, algorithmic thinking, research and communication skills).
Level II Content

Introduces 10 fundamental areas of computer science:

- computer organization,
- algorithmic problem solving,
- networks,
- applications (e.g., web page design),
- hierarchy and abstraction,
- computer logic/computer mathematics
- programming and software,
- modeling,
- ethics,
- careers
Level II Outcomes

Students should understand the following concepts:

- input, output, memory, storage, processing
- design, coding, testing and verification
- servers, queues, protocols, fault tolerance,
- buttons, menus, text areas, graphics, hypermedia
- binary numbers, logic, sets, and functions
- the broad utility of algorithmic problem solving
- compilers, interpreters, machine language
- security, privacy intellectual property, source reliability
- different careers in computing
Level III - Computer Science as Analysis and Design (Grade 10 or 11)

- Goal: to continue the study of computer science, placing particular emphasis on its features as a scientific and engineering discipline.
- This is a laboratory course
- Audience: (this is an elective course)
  - Students who enjoy science and math
  - Students who are preparing for college or IT careers
Level III Content

- Principles of design and problem solving, including style, abstraction, correctness and efficiency
- Simple data structures and their uses
- Design for usability - Web page design, interaction
- Fundamentals of hardware design
- Characteristics of compilers and operating systems
- What is a computationally "hard" problem? (e.g., ocean modeling, air traffic control, gene mapping)
- Unsolvable problems (e.g., the halting problem)
- Software engineering projects, teams, life cycle
- Social issues: intellectual property, professional practice
- Careers in computer science and IT
Level III Laboratory Work

- Design and implement algorithms that solve a variety of computational problems. Master:
  - Methods (functions) and parameters
  - Recursion
  - Objects and classes
  - Graphical and event-driven programming
- Logic and hardware design: gates and circuits, binary arithmetic, machine and assembly language
- Software requirements, design, teams, testing, documentation, tools
Level IV - Topics in Computer Science
(Grade 11 or 12)

- Elective courses for qualified and interested students
- Prerequisite = the Level II course
- Goals:
  - to provide an opportunity to explore special topics in greater depth
  - to prepare students for the IT workforce or for college
- School districts may offer any of the following:
  - Advanced Placement (AP) Computer Science
  - A projects-based course
  - A vendor-supplied course (maybe with certification)
Example Level IV Projects-Based Courses

- Desktop Publishing
- Presentation Design
- Multimedia
- Graphics
- Design and Development of Web Pages
- Web Programming
- Emerging Technologies (e.g., wireless)
- The Computer and Animation
- Network Design
- Programming simulations (e.g., ocean modeling)
Level IV - Example courses leading to industry certification

- A+ Certified Technician
  http://www.comptia.org/certification/a/default.asp

- Certified Internet Webmaster (CIW)
  http://www.ciwcertified.com/program/about.asp?comm=home&llm=1

- i-Net+
  http://www.computer-certification-training.com/CompTIA/inet/i-net.html
Next Steps: Implementation Challenges

- Curriculum Materials Development
- Teacher Preparation
- State-level Content Standards
- ASCD Buy-In
- Community and School District Support
Some References and Links


ISTE, National Educational Technology Standards for Teachers www.iste.org

National Research Council Committee on Information Technology Literacy, *Being Fluent with Information Technology* http://www.nap.edu/catalog/6482.html

Program for Initial Preparation of Teachers http://www.ncate.org/standard/programstds.htm