Bringing Computational Thinking to K-12

continued

TABLE 1: CORE COMPUTATIONAL THINKING CONCEPTS AND CAPABILITIES

CT Concept, Capability	CS	Math	Science	Social Studies	Language Arts
Data collection	Find a data source for a problem area	Find a data source for a problem area, for example, flipping coins or throwing dice	Collect data from an experiment	Study battle statistics or population data	Do linguistic analysis of sentences
Data analysis	Write a program to do basic statistical calculations on a set of data	Count occurrences of flips, dice throws and analyzing results	Analyze data from an experiment	Identify trends in data from statistics	Identify patterns for different sentence types
Data representation	Use data structures such as array, linked list, stack, queue, graph, hash table, etc.	Use histogram, pie chart, bar chart to represent data; use sets, lists, graphs, etc. To contain data	Summarize data from an experiment	Summarize and represent trends	Represent patterns of different sentence types
Problem Decomposition	Define objects and methods; define main and functions	Apply order of operations in an expression	Do a species classification		Write an outline
Abstraction	Use procedures to encapsulate a set of often repeated commands that perform a function; use conditionals, loops, recursion, etc.	Use variables in algebra; identify essential facts in a word problem; study functions in algebra compared to functions in programming; Use iteration to solve word problems	Build a model of a physical entity	Summarize facts; deduce conclusions from facts	Use of simile and metaphor; write a story with branches
Algorithms & procedures	Study classic algorithms; implement an algorithm for a problem area	Do long division, factoring; do carries in addition or subtraction	Do an experimental procedure		Write instructions
Automation		Use tools such as: geometer sketch pad; star logo; python code snippets	Use probeware	Use excel	Use a spell checker
Parallelization	Threading, pipelining, dividing up data or task in such a way to be processed in parallel	Solve linear systems; do matrix multiplication	Simultaneously run experiments with different parameters		
Simulation	Algorithm animation, parameter sweeping	Graph a function in a Cartesian plane and modify values of the variables	Simulate movement of the solar system	Play age of empires; Oregon trail	Do a re-enactment from a story

In attempting to define a classroom culture that would be most conducive to computational thinking, the participants identified strategies or characteristics that could be considered broadly beneficial to any learning experience. These included:

- Increased use by both teachers and students of computational vocabulary where appropriate to describe problems and solutions;
- Acceptance by both teachers and students of failed solution attempts, recognizing that early failure can often put you on the path to a successful outcome;
- Team work by students, with explicit use of:
  - *decomposition* - breaking problems down into smaller parts that may be more easily solved,
  - *abstraction* - simplifying from the concrete to the general as solutions are developed,
  - *negotiation* - groups within the team working together to merge parts of the solution into the whole, and
  - *consensus building* - working to build group solidarity behind one idea or solution.