Computational Thinking in K-12 Education

Chris Mayfield
mayfiecs@jmu.edu
Project CS4EDU

http://cs4edu.cs.purdue.edu/

Objective

• Create new pathways for education majors to become computationally educated secondary teachers

Highlights

• CS teaching endorsement (supplemental licensure)
• Computational thinking modules (and WebQuest)

New Courses

• Contemporary Issues in Computing
• Methods of Teaching Computer Science
Our Work with Ed Majors

Short Term Goal
• Prepare future educators from all subject areas to integrate computational thinking in their teaching

Long Term Goal
• K-12 students will have greater exposure to CS

Our Approach
• Develop new modules on CT (in class & web based)
• Embed in required courses for all education majors
• Survey the students before/after taking the modules

(Yadav et al., SIGCSE 2011)
What is Computational Thinking?

- Approach to problem solving using concepts from CS
- Fundamental skill for everyone [Wing, CACM 2006]
Informal Definition of CT

Informally, computational thinking describes the mental activity in formulating a problem to admit a computational solution. The solution can be carried out by a human or machine, or more generally, by combinations of humans and machines.

Jeannette M. Wing
see “The Link” CMU Newsletter, Spring 2011
Operational Definition of CT

CT is a problem-solving process that includes:

• Formulating problems [to enable computation]
• Logically organizing and analyzing data
• Representing data through abstractions
• Automating solutions through algorithmic thinking
• Identifying, analyzing, and implementing possible solutions
• Generalizing and transferring this problem solving process

http://csta.acm.org/Curriculum/sub/CompThinking.html
Daily Examples of CT

- Looking up a name in an alphabetically sorted list (*Binary Search*)
  - e.g., 100 names per page in list of 150,000 names
  - How to minimize the number of pages to look at?

- You and a friend are buying tickets for a movie (*Parallel Processing*)
  - There are 3 independent lines
  - How do you get your tickets ASAP?
Important Concepts of Computational Thinking
CT Concept: Abstraction

- **Decomposition**
  - Reformulating a seemingly difficult problem into one we know how to solve

- **Abstraction**
  - Pulling out the important details
  - Identifying principles that apply to other situations
Which of the following is NOT like the others?

[A] People standing in line at the store

[B] List of print jobs waiting to be printed

[C] Set of tennis balls in their container

[D] Vehicles lined up behind a toll booth

[E] Patients waiting to see the doctor
Abstraction: Queues vs Stacks

[A] People standing in line at the store (queue)
[B] List of print jobs waiting to be printed (queue)
[C] Set of tennis balls in their container (stack)
[D] Vehicles lined up behind a toll booth (queue)
[E] Patients waiting to see the doctor (queue)
Which of the following is NOT like the others?

[A] Files and directories on a hard disk.

[B] Parents and children in a pedigree chart.

[C] Brackets in the NCAA basketball tournament.

[D] My closest friends on Facebook / Twitter.

[E] The format of XML or PDF documents.
Abstraction: Graphs vs Trees

[A] Files and directories on a hard disk (tree)
[B] Parents and children in a pedigree chart (tree)
[C] Brackets in the NCAA basketball tournament (tree)
[D] My closest friends on Facebook / Twitter (graph)
[E] The format of XML or PDF documents (tree)
CT Concept: Algorithms

- Start
- Do I want to do this?
  - NO: Don't Do It
  - YES: Will it likely end in disaster?
    - NO: Will it make a good story anyway?
      - NO: Stop
      - YES: Do It
    - YES: Do It
Algorithm for a PB&J Sandwich

Materials:
• A jar of peanut butter
• A jar of jelly
• A loaf of sliced bread
• One butter knife

Your Task:
• What are the steps to make a peanut butter and jelly sandwich?
Other Daily Examples  (by Jeanette Wing)

- Putting things in your child’s knapsack for the day
  - Pre-fetching and caching
- Taking your kids to soccer, gymnastics, and swim practice
  - Traveling salesman (with more constraints)
- Cooking a gourmet meal
  - Parallel processing: You don’t want the meat to get cold while you’re cooking the vegetables
- Cleaning out your garage
  - Garbage collection: Keeping only what you need vs. throwing out stuff when you run out of space
- Storing away your child’s Lego pieces scattered on the LR floor
  - Using hashing (e.g., by shape, by color)
- Doing laundry, getting food at a buffet
  - Pipelining the wash, dry, and iron stages; plates, salad, entrée, dessert stations
- Even in grade school, we learn algorithms (long division, factoring, GCD, ...) and abstract data types (sets, tables, ...)
Benefits of Computational Thinking

• Moves students beyond technology literacy

• Creates problem solvers instead of software users

• Emphasizes creating knowledge and designing processes that can be automated

• Encourages creativity and problem solving

• Enhances many of the problem-solving techniques you already know and teach

(Source: Pat Phillips, NECC 2007, Atlanta)
Resources for Teaching Computational Thinking
http://csta.acm.org/ (membership is FREE!)

- Teaching and Learning Materials
  - Submitted by teachers, for teachers

- Brochures, Posters, Videos, Podcasts

- **ACM K-12 Model Curriculum**
  - Detailed lesson plans and activities (K-8, 9-10, 11-12)

- Professional Development (conferences/workshops)
• **Computational Thinking Toolkit**
  – PowerPoint presentations
  – Handouts and activities
  – Vocabulary & Progression Chart

• Video clips, online animations

• Implementation Strategies
  – Administration, counselors
  – Teachers, students, parents

http://www.iste.org/learn/computational-thinking.aspx
http://csunplugged.org/
(from University of Canterbury, New Zealand)

• **Kinesthetic activities**
  o including lesson plans and handouts

• **Video demonstrations:**
  http://www.youtube.com/csunplugged

• Great for motivating CT concepts
http://cs4fn.org/
(from Queen Mary University of London)

- **Magazines and Magic Books** (PDF format)
- Interactive applets for computational thinking
  - e.g., http://cs4fn.org/algorithms/swappuzzle/
  - Try to design an algorithm for any size board

- Lesson plans for in-class group activities
Google's CT Repository

http://www.google.com/edu/computational-thinking/

- **Lessons and Examples** (Math & Science)
  - "Easily incorporate CT into your curriculum"

- Online discussion forums

- Introduction to Python programming
  - Very simple; designed for 6th grade and up
Computer Science Education Week (December 9-15, 2012) is a call to action to:

- share information and offer activities
- advocate for computing
- elevate computer science education
- for students at all levels