A Model Curriculum for K-12 Computer Science

HI!

Luke Satoru Gusukuma

Luke Gusukuma: Composer, Computer Scientist, Student

- Virginia Tech 2nd Computer Science Masters Student
- Undergraduate Degrees in Music (Flute Performance) and Computer Science (from Tech)
- Hobbies: Swing Dance, Ballroom Dance, Flute, Origami
- Half Okinawan, half Filipino

Summary

- Tucker, Allen, et al. "A model curriculum for K-12 computer science: Final report of the ACM K-12 task force curriculum committee." *ACM, New York* (2003).
- This report, proposes a 4 level K-12 curriculum for Computer Science, via integrating CS principles into the curriculum from K-910, and offering elective courses at the high school level. The report first summarizes the current state of Computer Science Education in public school systems in the US, comparing it with other school systems in other countries, and then brings up the term IT Fluency and its distinction from CS, and then goes into the details of the proposed 4 level curriculum. The four levels are outlined as follows: the first level is to establish the foundations of CS; the second level informs students about CS's place in the modern world; the third level explores Analysis and design within CS; and the fourth level covers specialized topics in Computer Science. In an appendix to the report, there is a list of several samples of what activities that would be part of this curriculum might be.

The Four Level framework – Summary Point

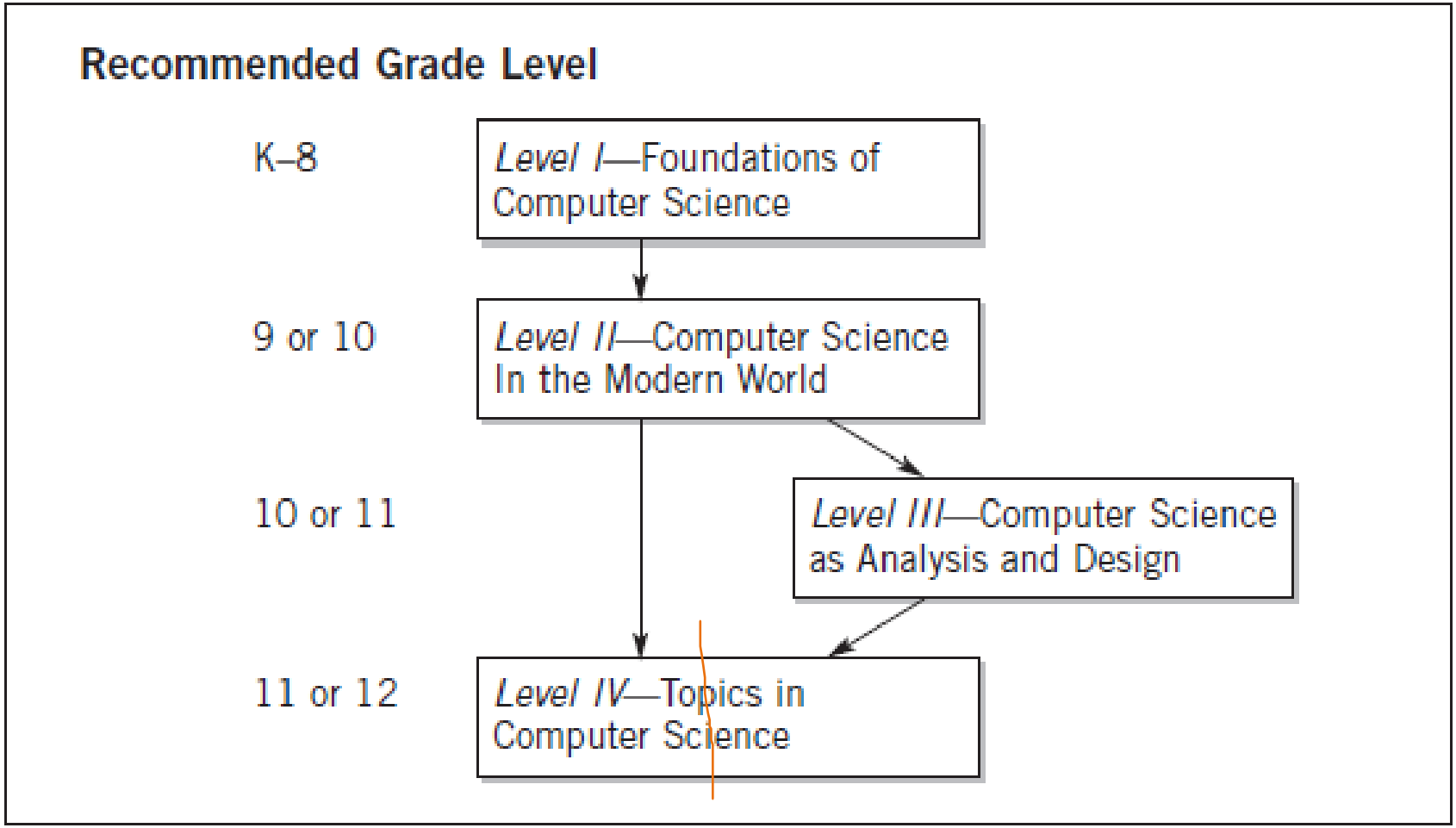

Figure 1. Structure of a K–12 Computer Science Curriculum

The Four Level framework – Summary Point

- The first two levels should be taught to everyone
- The last two levels are for those with interests in computer science

The Four Level framework: Level One – Summary Point

- Foundations of Computer Science, K-8
- Foundational concepts in CS should be taught by
  - Integrating basic skills in technology with simple ideas about algorithmic thinking
  - Adding short modules to
    - Science
    - Mathematics
    - Social studies

The Four Level framework: Level One – Summary Point

- Out of middle school:
  - “students should have gained experience using computers … They should have used, modified, and created files for a variety of purposes, accessed the Internet and databases for both research and communication, and used other tools such as spreadsheets and graphics. Finally, they should have been introduced to the basic idea of algorithmic thinking and its uses in their daily lives.”

The Four Level framework: Level One – Discussion Point

Activity: . . . . . Color by Numbers

Time: . . . . . 3 hours

Description: . . . . . The computer stores drawings, photographs, text, and other pictures using only numbers. This activity demonstrates how that is done.

Level: . . . . . I (K–2)

Topics: . . . . . 11—using 0s and 1s to represent information

Prior Knowledge: . . . . . Grade 2 geometry (exploring shapes), counting, graphing

Planning Notes:

- Motivational discussion questions include, "What does a fax machine do?"
- "In what situations would a computer want to store pictures?"
- "How do computers store pictures when they can only use numbers?"

Teaching/Learning Strategies:

- A 5x6 rectangular grid is used as a basis for representing different images (such as letters) by coloring in some of the squares (pixels).
- Coding of the image is done by scanning the sequences of 1s (shaded squares) and 0s in each row of the grid and recording the length of each sequence.

Assessment and Evaluation: . . . . . Worksheet activities.

Accommodations: . . . . . No computers are required; students use two worksheet activities, called "Kid fax" and "Make your own picture"

Resources: . . . . . See *www.unplugged.canterbury.ac.nz* to learn more about this activity.

The Four Level framework: Level Two – Summary Point

- Computer Science in the modern world
- one-year course accessible to all students, college bound or not
- "In this course, high school students can acquire a fundamental understanding of the operation of computers and computer networks and create useful programs implementing simple algorithms. By developing Web pages that include images, sound, and text, they can acquire a working understanding of the Internet, common formats for data transmission, and some insights into the design of the human computer interface. Exposure to career possibilities and discussion of ethical issues relating to computers should also be important threads in this course."

The Four Level framework: Level Two – Summary Point

- This course provides the first opportunity to view computer science as a coherent field of study and professional engagement. That is, while IT fluency focuses on technological skills and their uses in other academic subjects, this course is a study of computer science as an academic subject per se.

The Four Level framework: Level Two – Summary Point

- computer organization and the major components (input, output, memory, storage, processing, software, operating system, etc.)
- basic steps in algorithmic problem solving
- basic components of computer networks
- Organization of Internet elements, Web page design and hypermedia
- The notion of hierarchy and abstraction in computing, including high-level languages, translation (compilers, interpreters, linking), machine languages, instruction sets, and logic circuits.
- connection between mathematics and computer science, including binary numbers, logic, sets, and functions.
- notion of computers as models of intelligent behavior (e.g. robot motion, speech recognition, and computer vision), and what distinguishes humans from machines.
- Examples that identify the broad interdisciplinary utility of computers and algorithmic problem solving in the modern world.
- Ethical issues that relate to computers and networks (e.g. security, privacy, intellectual property, reliability) and the positive and negative impact of technology on human culture.
- Identification of careers in computing and their connection with the subjects studied in this course

The Four Level framework: Level Two – Discussion Point

- computer organization and the major components (input, output, memory, storage, processing, software, operating system, etc.)
- basic steps in algorithmic problem solving
- basic components of computer networks
- Organization of Internet elements, Web page design and hypermedia
- The notion of hierarchy and abstraction in computing, including high-level languages, translation (compilers, interpreters, linking), machine languages, instruction sets, and logic circuits
- connection between mathematics and computer science, including binary numbers, logic, sets, and functions.
- notion of computers as models of intelligent behavior (e.g. robot motion, speech recognition, and computer vision), and what distinguishes humans from machines.
- Examples that identify the broad interdisciplinary utility of computers and algorithmic problem solving in the modern world.
- Ethical issues that relate to computers and networks (e.g. security, privacy, intellectual property, reliability) and the positive and negative impact of technology on human culture.
- Identification of careers in computing and their connection with the subjects studied in this course

The Four Level framework: Level Two – Discussion Point

- Do they really need to know about computer organization?
- How deep into Computer Science do we want to go?
- How much detail of Computer Science is really necessary to teach computational thinking?

The Four Level framework: Level Three – Summary Point

- "The major goal of this course is for students to develop the computer science skills of algorithm development, problem solving, and programming while using software engineering principles. While the emphasis of the course will be on programming, students will also be introduced to other important topics, such as interface design, the limits of computers, and societal and ethical issues of software engineering."

The Four Level framework: Level Four – Summary Point

- "At this level, interested and qualified students should be able to select one from among several electives to gain depth of understanding or special skills in particular areas of computer science. All of these electives will require the Level II course as a prerequisite, while some may require the Level III course as well. Most important, these courses provide students with an opportunity to explore topics of personal interest in greater depth, and thus prepare for the workplace or for further study at the post-secondary level."

Why teach Computational Thinking?

How far should we go?

Computer Science is Important Intellectually – Summary Point

- "Computers are fundamentally different from other technological inventions in the past in that they directly augment human thought, rather than, say, the functions of our muscles or our senses."
- "An engineer using a computer to design a bridge must understand how the maximum capacity estimates were computed and how reliable they are."
- "An educated citizen using a voting machine or bidding in an eBay auction should have a basic understanding of the underlying algorithms of such conveniences, as well as the security and privacy issues that arise when information is transmitted and stored digitally."
- "Computer science students learn logical reasoning, algorithmic thinking, design and structured problem solving—all concepts and skills that are valuable well beyond the computer science classroom."
- "Students gain awareness of the resources required to implement and deploy a solution and how to deal with real-world constraints."

Computer Science is Important Intellectually – Discussion Point

- An engineer using a computer to design a bridge must understand how the maximum capacity estimates were computed and how reliable they are.
  - Engineers already learn and know the math. Reliability knowledge is important, but is that really computer Science?
- An educated citizen using a voting machine or bidding in an eBay auction should have a basic understanding of the underlying algorithms of such conveniences, as well as the security and privacy issues that arise when information is transmitted and stored digitally.
  - How is this different from understanding how any system works? People should know how legislation affects them and their rights, is this any different?
- Students gain awareness of the resources required to implement and deploy a solution and how to deal with real-world constraints.
  - Isn't this Engineering?
- Computer science students learn logical reasoning, algorithmic thinking, design and structured problem solving—all concepts and skills that are valuable well beyond the computer science classroom.
  - This is the focus of Computational Thinking, right?

Versatilists – Summary Point

- Thomas Friedman, in his best-selling book The World is Flat, (2006) argues that our economy most needs "Versatilists," people who have expertise in some domain and in technology. Computer science is the glue that makes it possible for these Versatilists to work together.

Versatilists – Discussion Point

- In what cases does a client actually need some domain knowledge of computer science?
- "Often a computer scientist works closely with business people, scientists, and other experts to understand the issues, and to define the problem so explicitly that it can be represented in a computer."
  - For the application of Computer Science, is it REALLY necessary for clients to have knowledge of Computer Science? Of Computational Thinking?
- Can't we just create an in-between class of "computational [insert subject matter here]" fields? Don't we kind of do that already?

IT Fluency – Summary Point

- "IT Literacy is the capability to use today's technology in one's own field, the notion of IT Fluency adds the capability to independently learn and use new technology as it evolves throughout one's professional lifetime.“
- “Moreover, IT fluency also includes the active use of algorithmic thinking (including programming) to solve problems, whereas IT literacy is more limited in scope."

IT Fluency – Summary Point

- “the complexity of algorithms is a fundamental idea in computer science but would probably not appear in an IT curriculum. While IT is an applied field of study, driven by the practical benefits of its knowledge, computer science has scientific and mathematical, as well as practical, dimensions.”
- "The idea of IT fluency (National Research Council, 1999) was proposed as a minimum standard that all college students should achieve by the time they graduate. A "fluent" graduate would master IT on three orthogonal axes — concepts, capabilities, and skills."

IT Fluency (The Laundry list) – Summary Point

- "Concepts are the 10 basic ideas that underlie modern computers, networks, and information:
  - Computer organization
  - information systems
  - networks
  - digital representation of information
  - information organization
  - modeling and abstraction
  - algorithmic thinking and programming
  - universality
  - limitations of information technology
  - and societal impact of information technology."

IT Fluency (The Laundry list) – Summary Point

- "Capabilities are the 10 fundamental abilities for using IT to solve a problem:
  - Engage in sustained reasoning
  - manage complexity
  - test a solution
  - manage faulty systems and software
  - organize and navigate information structures and evaluate information
  - collaborate
  - communicate to other audiences
  - expect the unexpected
  - anticipate changing technologies
  - and think abstractly about IT."

IT Fluency (The Laundry list) – Summary Point

- "Skills are the 10 abilities to use today's computer applications in one's own work:
  - Set up a personal computer
  - use basic operating system features
  - use a word processor and create a document
  - use a graphics or artwork package to create illustrations, slides, and images
  - connect a computer to a network
  - use the Internet to find information and resources
  - use a computer to communicate with others
  - use a spreadsheet to model simple processes or financial tables
  - use a database system to set up and access information
  - and use instructional materials to learn about new applications or features."

IT Fluency – Discussion Point

- Concepts
  - **Modeling** and **abstraction**
  - **Algorithmic thinking** and programming
  - universality
  - **limitations** of information technology
  - **societal impact** of information technology

- Capabilities
  - organize and navigate **information** structures and evaluate information
  - Engage in sustained **reasoning**

General Discussion Points

- What parts of computational thinking do we actually want to be part of a general curriculum?
- What parts of computational thinking are actually necessary to have by the populous.
- How do we integrate this into the current curriculum?
- Should our focus be on integrating and working with IT Fluency, as defined by the National Research Council?