Computer Science for All i2i Guide for Businesses

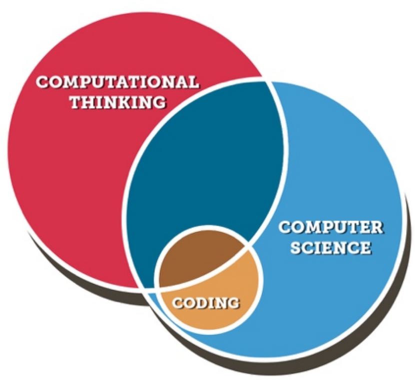
Purpose:

The purpose of this guide is to assist in making visible key ideas, content, and enduring practices and habits of mind related to Computer Science (CS) and Computational Thinking (CT) and their relationship to the proficiencies necessary for success in current and future businesses and occupations. By making these competencies explicit in the examples you present from your particular business sector, educators can link to real life examples as they prepare learners for these types of future opportunities in the workforce. Educators and educational leaders can also share your specific examples with parents, families, and the broader community. Increasing the visibility and importance of Computer Science and Computational Thinking learning will enhance both educational efforts and support the development of the future workforce.

Definitions (See additional handout):

There are numerous definitions of Computer Science and Computations Thinking in the CS/CT community. Review the two definitions are below and those included in the supplemental handout.

- **"Computational thinking** involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to **computer science."**
- **Computer Science** is the study and application of problem solving and design through computational thinking, programming, analysis, and creative modeling using computers, hardware, software, and algorithmic processes in order to solve real world problems, be ethically responsible, improve efficiency, and increase access to knowledge.



In both definitions above, and those on the associated handouts, it is important to note that CS/CT is not limited just to programming and coding computers. CS/CT is inclusive of a range of skills, knowledge, and practices on and off computers.

Practices and Habits of Mind Embedded In CS/CT

In order to be successful in the workplace we need a citizenry who are able to make sense of problems and persevere in solving them, manage and understand data, use and design technology solutions, and communicate information among other enduring habits of mind. These practices are embedded within activities and learning of CS/CT as well as other Science, Technology, Engineering, and Mathematics (STEM) domains. (See examples in the enduring practices handout).

Guiding questions:

1. When you look at the CS/CT definitions and practices, where within your business do you see applications of these competencies?

2. What are specific examples where these practices are currently needed?

3. Where within your industry do you see future needs for these types of competencies?

4. Do you have suggestions for integrating CS to support enduring practices? If so, what are they?

5. In what ways do you envision educators could engage with you or your business to mutually support the development of critical CS/CT proficiencies and/or deepen learners' and community members' knowledge of how these skills connect to real world careers?

TASK:

Share examples from your business of where CS/CT knowledge and practices are currently being used. Include examples of future opportunities to apply these proficiencies based on where you see your industry heading.

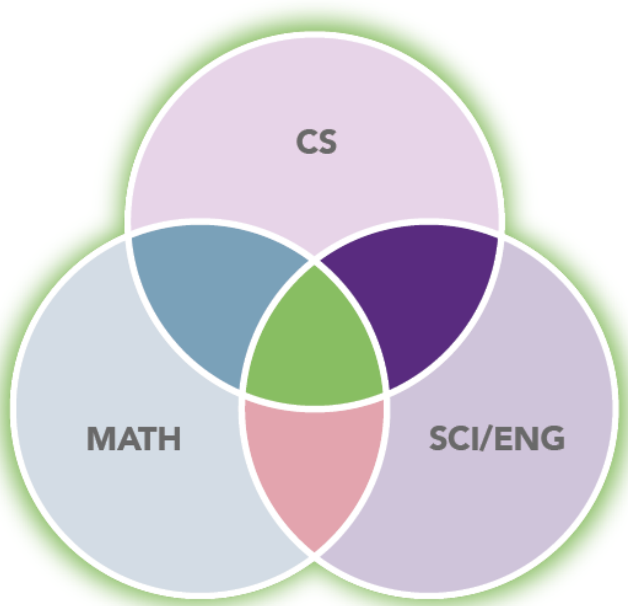
Practices and Proficiencies	Examples

NOTE: We'd love for you to make relevant connections to computational thinking, problem solving and collaborative components of the daily work that you do that utilizes or connects to computer science in some way.

Source: https://k12cs.org/navigating-the-practices/

Enduring Practices[1]:

RELATIONSHIPS BETWEEN COMPUTER SCIENCE, SCIENCE AND ENGINEERING, AND MATH PRACTICES



CS + Math

- **Develop and use abstractions**
  M2. Reason abstractly and quantitatively
  M7. Look for and make use of structure
  M8. Look for and express regularity in repeated reasoning
  CS4. Developing and Using Abstractions
- **Use tools when collaborating**
  M5. Use appropriate tools strategically
  CS2. Collaborating Around Computing
- **Communicate precisely**
  M6. Attend to precision
  CS7. Communicating About Computing

CS + Sci/Eng

- **Communicate with data**
  S4. Analyze and interpret data
  CS7. Communicating About Computing
- **Create artifacts**
  S3. Plan and carry out investigations
  S6. Construct explanations and design solutions
  CS4. Developing and Using Abstractions
  CS5. Creating Computa-tional Artifacts
  CS6. Testing and Refining Computational Artifacts

CS + Math + Sci/Eng

- **Model**
  S2. Develop and use models
  M4. Model with mathematics
  CS4. Developing and Using Abstractions
  CS6. Testing and Refining Computational Artifacts
- **Use computational thinking**
  S5. Use mathematics and computational thinking
  CS3. Recognizing and Defining Computational Problems
  CS4. Developing and Using Abstractions
  CS5. Creating Computational Artifacts
- **Define problems**
  S1. Ask questions and define problems
  M1. Make sense of problems and persevere in solving them
  CS3. Recognizing and Defining Computational Problems
- **Communicate rationale**
  S7. Engage in argument from evidence
  S8. Obtain, evaluate, and communicate information
  M3. Construct viable arguments and critique the reasoning of others
  CS7. Communicating About Computing

* Computer science practices also overlap with practices in other domains, including English language arts. For example, *CS1. Fostering an Inclusive Computing Culture* and *CS2. Collaborating Around Computing* overlap with *E7. Come to understand other perspectives and cultures through reading, listening, and collaborations.*

[1] Retrieved from the K12 Computer Science Framework https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf

What do the core computational thinking concepts and capabilities look like across the curricula?

According to Valerie Barr and Chris Stephenson (2011), "The process of increasing student exposure to computational thinking in K-12 is complex, requiring systemic change, teacher engagement, and development of significant resources. Collaboration with the computer science education community is vital to this effort." Equally as important are efforts within the business and higher education community.

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

From Barr & Stephenson (2011), p.52

Barr, V., Stephenson, C. (2011). "Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?" *ACM Inroads,* 2(1): 48-54.

Definitions of Computational Thinking and Computer Science

<table>
<tr>
<td>“Computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry out.”</td>
<td>“Current integration of computational thinking into the K-12 curriculum comes in two forms: in computer science classes directly or through the use and measure of computational thinking techniques in other subjects.”</td>
<td>“Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science.”</td>
</tr>
<tr>
<td>“Teachers in Science, Technology, Engineering, and Mathematics (STEM) focused classrooms that include computational thinking, allow students to practice problem-solving skills such as trial and error” (Barr, et al., 2011)</td>
<td>“Computational thinking (CT) tools permeate the way that scientists, engineers, and mathematicians conduct research and make sense of the world. CT has allowed for breakthroughs in sequencing the human genome, predicting hurricane paths, modeling black holes, and optimizing food pantry routes.”</td>
<td>“Computer science is the science that deals with the theory and methods of processing information in digital computers, the design of computer hardware and software, and the applications of computers.”</td>
</tr>
<tr>
<td>“Science and mathematics are becoming computational endeavors… Next Generation Science Standards…include “computational thinking” as a core scientific practice.”</td>
<td rowspan="3">Computational thinking is a problem-solving process that includes:
• Formulating problems in a way that enables us to use a computer and other tools to help solve them
• Logically organizing and analyzing data
• Representing data through abstractions such as models and simulations
• Automating solutions with the goal of achieving the most efficient and effective combination of steps and resources
• Generalizing and transferring this problem-solving process to a wide variety of problems</td>
<td>“Computer science is the study of the theory, experimentation, and engineering that form the basis for the design and use of computers.”</td>
</tr>
<tr>
<td rowspan="2">“Computational thinking (CT) is an approach to solving problems in a way that can be implemented with a computer. Students become not mere tool users but tool builders. They use a set of concepts, such as abstraction, recursion, and iteration, to process and analyze data, and to create real and virtual artifacts. CT is a problem solving methodologythat can be automated and transferred and applied across subjects.”</td>
<td>“Computer science is the study of the hardware, software, networking, and all the processes that fall under the umbrella of giving life to a machine to enable it to perform complicated tasks and actions.”</td>
</tr>
<tr>
<td>“Computer Science (CS) is about designing and developing computing systems to solve problems. It is a science, so it comprises a set of ideas and principles.”</td>
</tr>
</table>

Computational Thinking is…

- The process of solving problems that can be done by humans or machines*
- Step by step plan or procedures (sequence, selection, repetition)*
- Logical thinking*
- Analytical thinking skills*
- Logic puzzles
- Problem solving
- Daily life skills
- Changing Vocabulary*

*Note: NOT devices, machines, or hardware

References:

Barr, D., Harrison, J., Conery, L. (2011). "Computational thinking: A digital age skill for everyone." *Learning and Leading with Technology,* 38(6): 20-23.

Barr, V., Stephenson, C. (2011). "Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?" *ACM Inroads,* 2(1): 48-54.

Computational thinking. (2018, January 31). In *Wikipedia, The Free Encyclopedia*. Retrieved 02:57, February 8, 2018, from https://en.wikipedia.org/w/index.php?title=Computational_thinking&oldid=823238273

CT-STEM Project Investigators (2016). CT STEM. Retrieved from http://ct-stem.northwestern.edu/training/.

Grover, S., Pea, R., (2013) Computational Thinking in K–12 A Review of the State of the Field. Retrieved from http://journals.sagepub.com/doi/abs/10.3102/0013189X12463051

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. & Wilensky, U., (2016). "Defining Comutational Thinking for Mathematics and Science Classrooms." *Journal of Science Education and Technology,* 25: 127-147.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2015). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology* (1-21)

Wing, J. 2006. Computational Thinking. *Communications of the ACM, 49(3). https://www.cs.cmu.edu/15110-s13/Wing06-ct.pdf* Wing, 2006. Accessed February 7, 2018