ACCESS, INNOVATION, AND EVALUATION
2020 Education Innovation and Research (EIR) Project Directors and Evaluators Technical Assistance Meeting
Virtual Meeting | October 21–23, 2020

Measuring Computational Thinking and Computer Science Outcomes

Barbara Goodson | Maureen Sarna (Presenters)

Abt Associates

Barbara Goodson

Role: Principal Investigator, TA Leader & TA Liaison

Background: Barbara is a Principal Scientist in Abt Associates' Social and Economic Policy division. She has more than three decades of experience in designing and conducting experimental and quasi-experimental impact evaluations. She is the Principal Investigator overseeing the evaluation technical assistance team for the EIR and i3 grant programs, and was previously a First in the World grant program technical assistance liaison.

Maureen Sarna

Role: EIR Deputy Project Director (DPD)

Background: Maureen is an Associate in Abt Associates' Social and Economic Policy division. She joined the i3 evaluation technical assistance team in 2013 to conduct reviews of fidelity of implementation studies and has served as the Deputy Project Director for that team since 2017. Maureen is also the Deputy Project Director on the FY 2017 and FY 2018 EIR evaluation TA contracts. She has almost a decade of experience in evaluation research in the fields of fields of education, workforce development, and work-family policies.

Stephen Uzzo

Role: Grantee

Background: Stephen Uzzo is Chief Scientist for the New York Hall of Science and Adjuct Assistant Professor of Education for The New York Institute of Technology. He has been working for several decades to make computer science education more inclusive and equitable. Dr. Uzzo's background includes STEM teacher education and research practice partnerships, learning research and public experiences in science.

Darcy Ronan

Role: Grantee

Background: Darcy Ronan, Ph.D. is an Assistant Professor and Director of STE(A)M Education programs in the Farrington College of Education at Sacred Heart University in CT. A science educator and curriculum developer by background, she is a STEM+CS education researcher with a focus on development and impact of identities and beliefs in populations encountering STEM such as teachers, undergraduate researchers, and career aspirants.

Agenda

- EIR Priorities for STEM, CS, CT
- What Is Computational Thinking?
- Measuring Computational Thinking and Computing Skills
- Student and Teacher Attitudes & Beliefs About STEM, CT, and CS
- Leveraging EIR to Advance the Field

Terminology--edits

- STEM: Science, Technology, Engineering, and Mathematics
  - Combined content across these four specific disciplines taught in an interdisciplinary and applied approach
  - Emphasis on hands-on experiences that provide opportunities for students to gain and apply relevant, “real-world” knowledge in the classroom
  - Emphasis on 21st century skills — ability to solve problems, make sense of information, and know how to gather and evaluate evidence to make decisions

Terminology, Continued

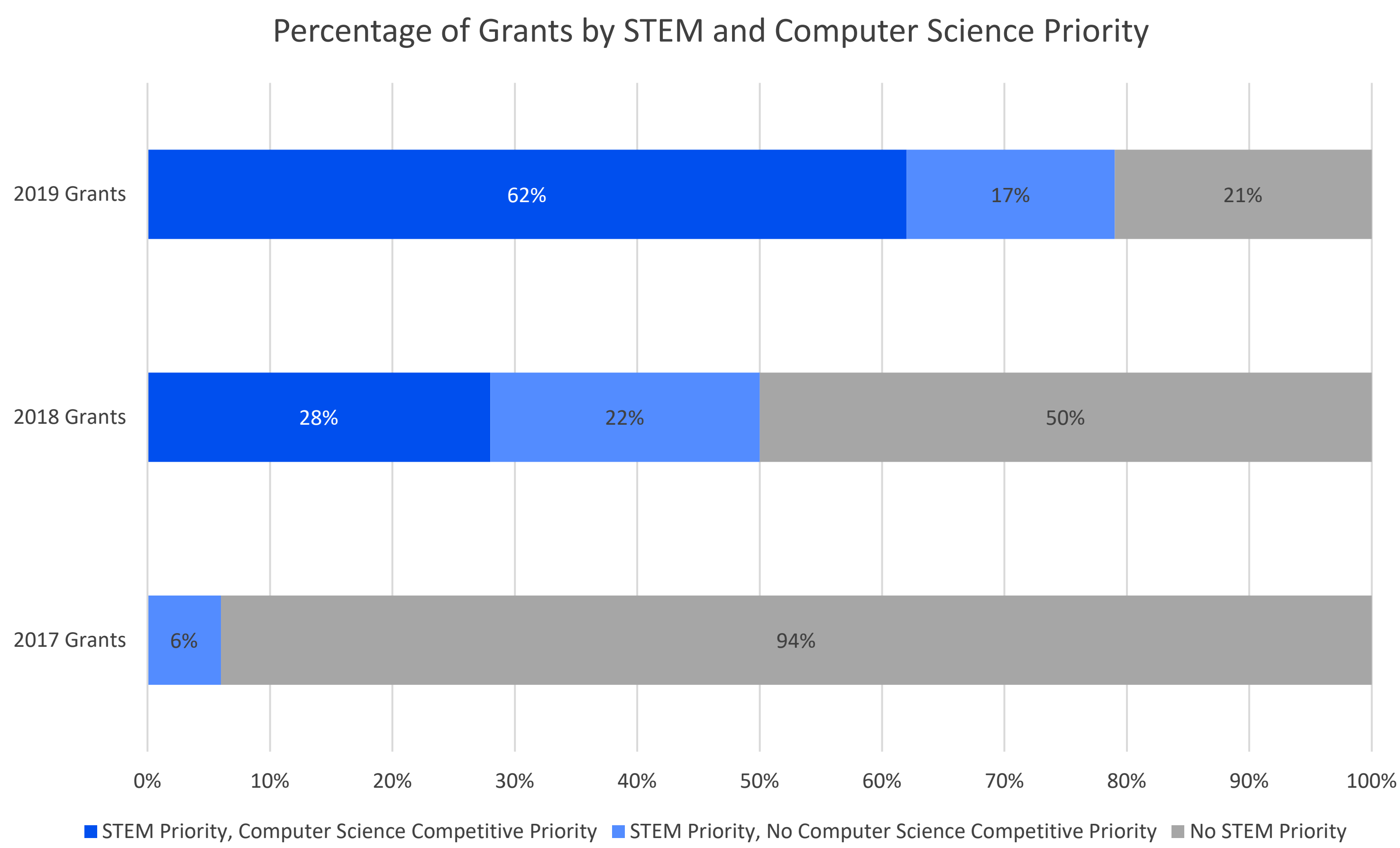
- CS: Computer Science
  - Discipline that spans a range of topics from theoretical studies of computation and information to the practical issues of implementing computing systems in hardware and software
- CT: Computational Thinking
  - The thought processes involved in expressing solutions as computational steps that may be carried out by a computer
- *New urgency for defining CT and its relationship to CS, to provide theoretical grounding for the form CT should take in science and mathematics classrooms*

EIR Priorities for STEM, CS, CT

EIR Priorities Reflect Federal Policy Around STEM

- Federal strategy for the next five years: All Americans will have lifelong access to high-quality STEM education so the US is the global leader in STEM literacy, innovation, and employment
- EIR reflects this federal vision in its priority for STEM innovations in FY2018 and FY2019 grant competitions
- EIR also prioritizes STEM innovations that develop computer science skills and positions computational thinking as part of computer science:
  - Computer science includes "*computing principles and theories, algorithmic processes, computational thinking, computer hardware, software design, coding, analytics, and computer applications*"

Increasing EIR Focus on STEM and Computer Science



Increased Need for New Measures of CS and CT

- Increased number of EIR STEM and CS interventions has also shifted the focus to measures of CS and CT skills as key student outcomes
- Increasing need for valid measures
- Only a few states conduct standardized testing of STEM knowledge and none assess CS skills
  - Using state math/science tests to assess STEM outcomes perceived as not well targeted and unlikely to show program effects
- Only one standardized test of CS skills exists: College Board Advanced Placement − Computer Science Principles exam
  - Assesses understanding of the computational thinking practices and learning objectives in course framework

What Is Computational Thinking?

13

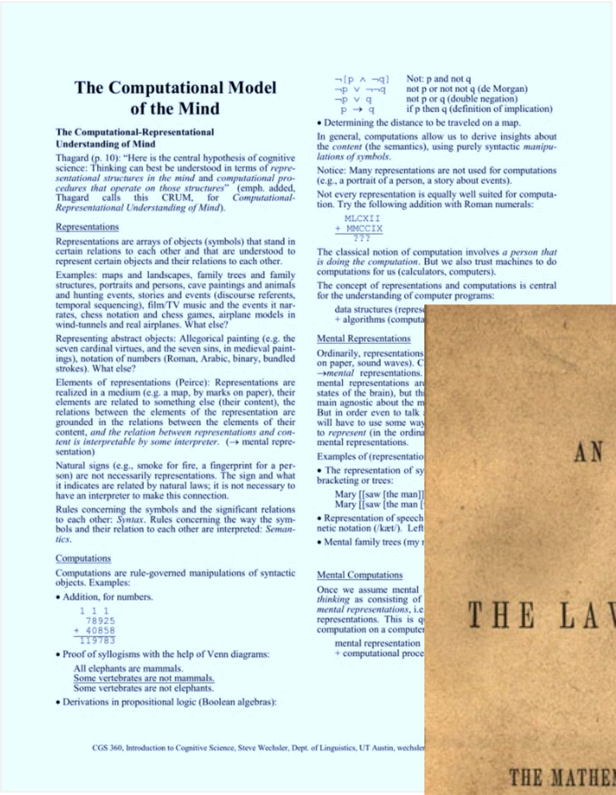
The Problem of Definitions

- Isn’t it just mathematical thinking?
- Isn’t it just trying to think like a computer?
- Isn’t it just the skills needed to do computer programming?
- Isn’t this just like critical thinking?
- Isn’t this just the same thing as an IQ test of logical thinking?

The Mind as Partly a Computer

P. Thagard. (2005). *Mind: Introduction to Cognitive Science.* The MIT Press.

G. Boole. (1854). *An Investigation of the Laws of Thought, On Which Are Founded the Mathematical Theories of Logic and Probabilities.* Walton and Maberly.



The Computational Model of the Mind

The Computational-Representational Understanding of Mind

Thagard (p. 10): "Here is the central hypothesis of cognitive science: Thinking can best be understood in terms of *representational structures in the mind* and *computational procedures that operate on those structures*" (emph. added, Thagard calls this CRUM, for *Computational-Representational Understanding of Mind*).

Representations

Representations are arrays of objects (symbols) that stand in certain relations to each other and that are understood to represent certain objects and their relations to each other.

Examples: maps and landscapes, family trees and family structures, portraits and persons, cave paintings and animals and hunting events, stories and events (discourse referents, temporal sequencing), film/TV music and the events it narrates, chess notation and chess games, airplane models in wind-tunnels and real airplanes. What else?

Representing abstract objects: Allegorical painting (e.g. the seven cardinal virtues, and the seven sins, in medieval paintings), notation of numbers (Roman, Arabic, binary, bundled strokes). What else?

Elements of representations (Peirce): Representations are realized in a medium (e.g. a map, by marks on paper), their elements are related to something else (their content), the relations between the elements of the representation are grounded in the relations between the elements of their content, *and the relation between representations and content is interpretable by some interpreter.* (→ mental representation)

Natural signs (e.g., smoke for fire, a fingerprint for a person) are not necessarily representations. The sign and what it indicates are related by natural laws; it is not necessary to have an interpreter to make this connection.

Rules concerning the symbols and the significant relations to each other: *Syntax*. Rules concerning the way the symbols and their relation to each other are interpreted: *Semantics.*

Computations

Computations are rule-governed manipulations of syntactic objects. Examples:

- Addition, for numbers.

```
  1 1 1
  78925
+ 40858
 119783
```

- Proof of syllogisms with the help of Venn diagrams:

All elephants are mammals.
Some vertebrates are not mammals.
Some vertebrates are not elephants.

- Derivations in propositional logic (Boolean algebras):

$\neg[p \wedge \neg q]$	Not: p and not q
$\neg p \vee \neg\neg q$	not p or not not q (de Morgan)
$\neg p \vee q$	not p or q (double negation)
$p \rightarrow q$	if p then q (definition of implication)

- Determining the distance to be traveled on a map.

In general, computations allow us to derive insights about the *content* (the semantics), using purely syntactic *manipulations of symbols.*

Notice: Many representations are not used for computations (e.g., a portrait of a person, a story about events).

Not every representation is equally well suited for computation. Try the following addition with Roman numerals:

```
  MLCXII
+ MMCCIX
   ???
```

The classical notion of computation involves *a person that is doing the computation*. But we also trust machines to do computations for us (calculators, computers).

The concept of representations and computations is central for the understanding of computer programs:

data structures (represe
+ algorithms (computa

Mental Representations

Ordinarily, representations
on paper, sound waves). C
→*mental* representations.
mental representations ar
states of the brain), but th
main agnostic about the m
But in order even to talk
will have to use some way
to *represent* (in the ordina
mental representations.

Examples of (representatio

- The representation of sy
bracketing or trees:

Mary [[saw [the man]]
Mary [[saw [the man [

- Representation of speech
netic notation (/kæt/). Left

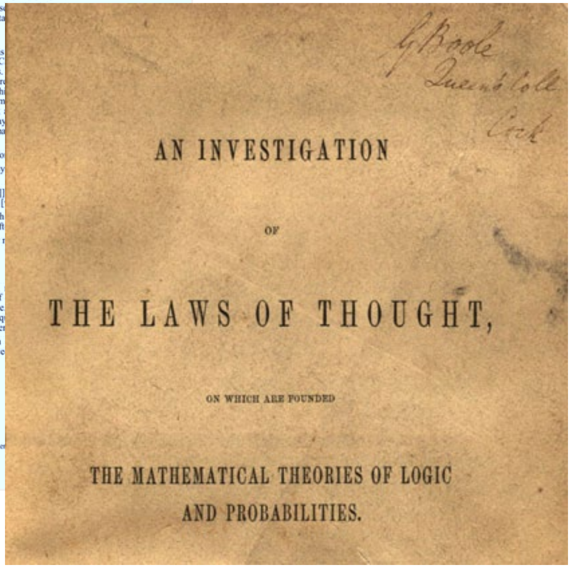
- Mental family trees (my

Mental Computations

Once we assume mental
thinking as consisting of
mental representations, i.e
representations. This is q
computation on a computer

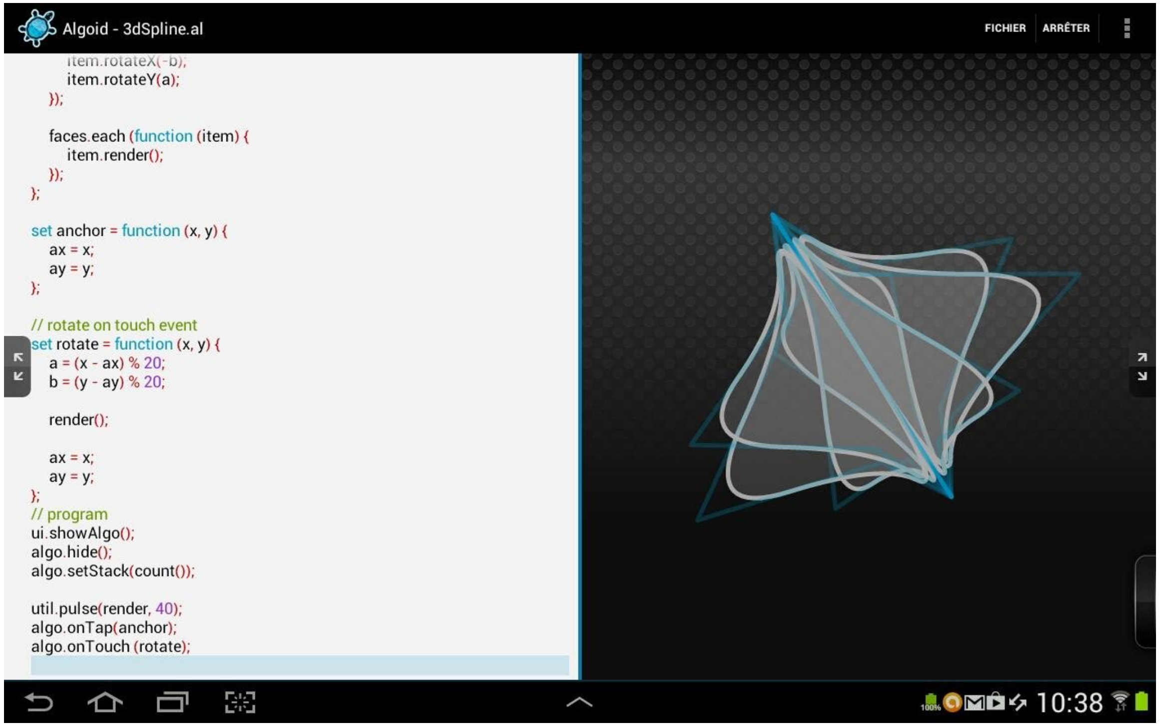
mental representation
+ computational proce

CGS 360, Introduction to Cognitive Science, Steve Wechsler, Dept. of Linguistics, UT Austin, wechsler



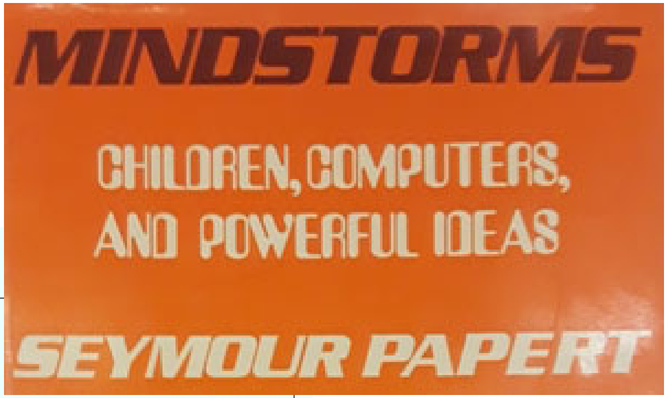
The Computer as Part of the Mind

Body-Syntonic Reasoning:
“Constructionism”

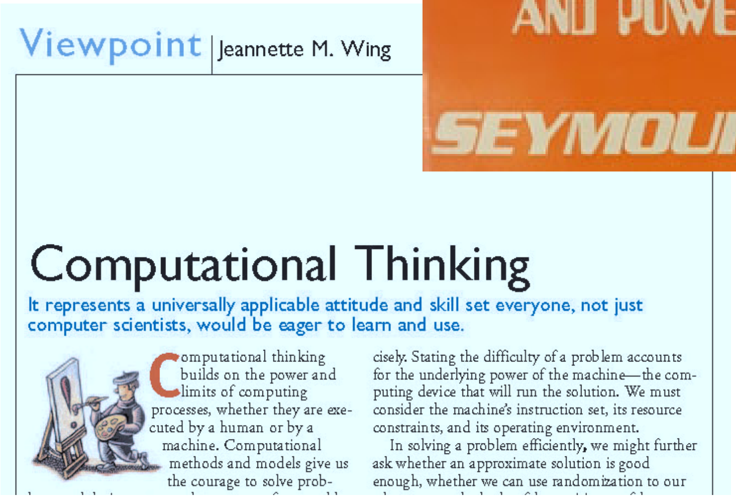


Computational Thinking as Epistemology

S. Papert. (1993). *Mindstorms: Children, Computers, and Powerful Ideas*. 2nd ed. Basic Books.



J. M. Wing. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.


Viewpoint | Jeannette M. Wing

Computational Thinking

It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.

Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine. Computational methods and models give us the courage to solve prob-

cisely. Stating the difficulty of a problem accounts for the underlying power of the machine—the computing device that will run the solution. We must consider the machine's instruction set, its resource constraints, and its operating environment.

In solving a problem efficiently, we might further ask whether an approximate solution is good enough, whether we can use randomization to our

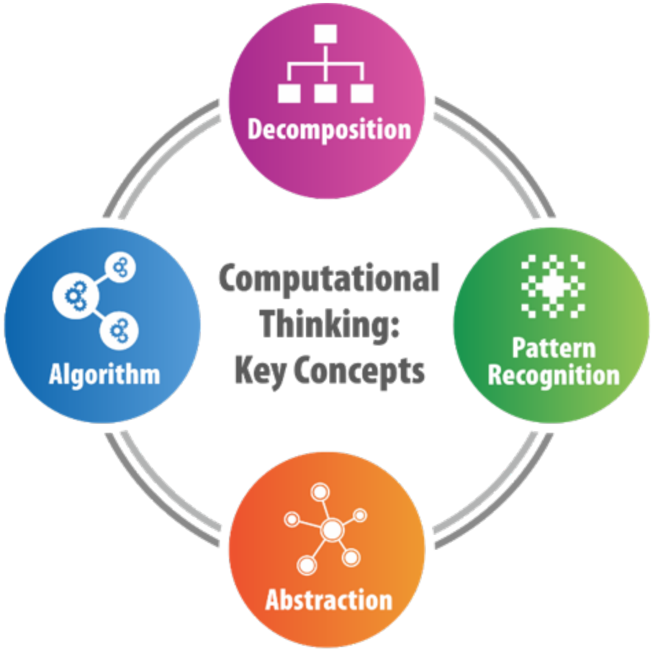
Basic Concepts in Computational Thinking

Decomposition — breaking down a complex problem or system into smaller, more manageable parts

Pattern Recognition — looking for similarities among and within problems

Abstraction — focusing on the important information only, ignoring irrelevant detail

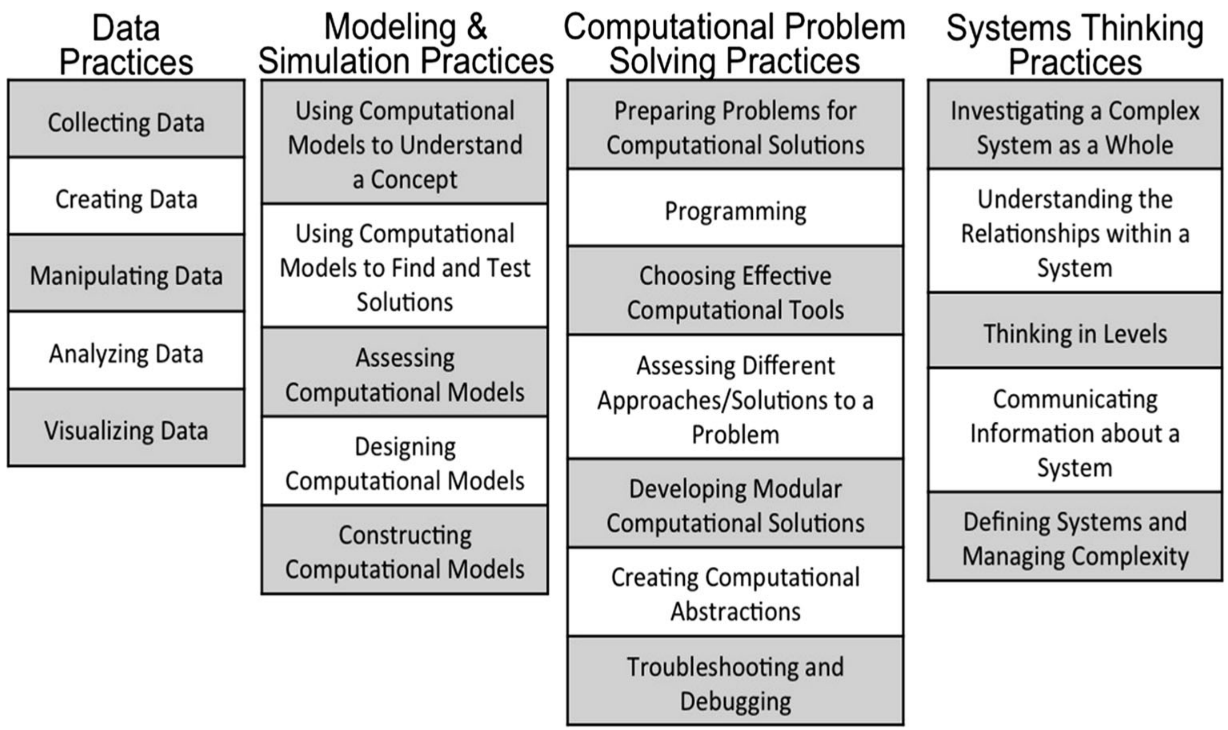
Algorithm — developing a step-by-step solution to the problem, or the rules to follow to solve the problem



Shah, V. 2018, CSpathshala.

Computational Thinking in STEM

D. Weintrop, E. Beheshti, M. Horn, K. Orton, K. Jona, L. Trouille, & U. Wilensky. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127-147.



Measuring Computational Thinking and Computer Science Proficiency

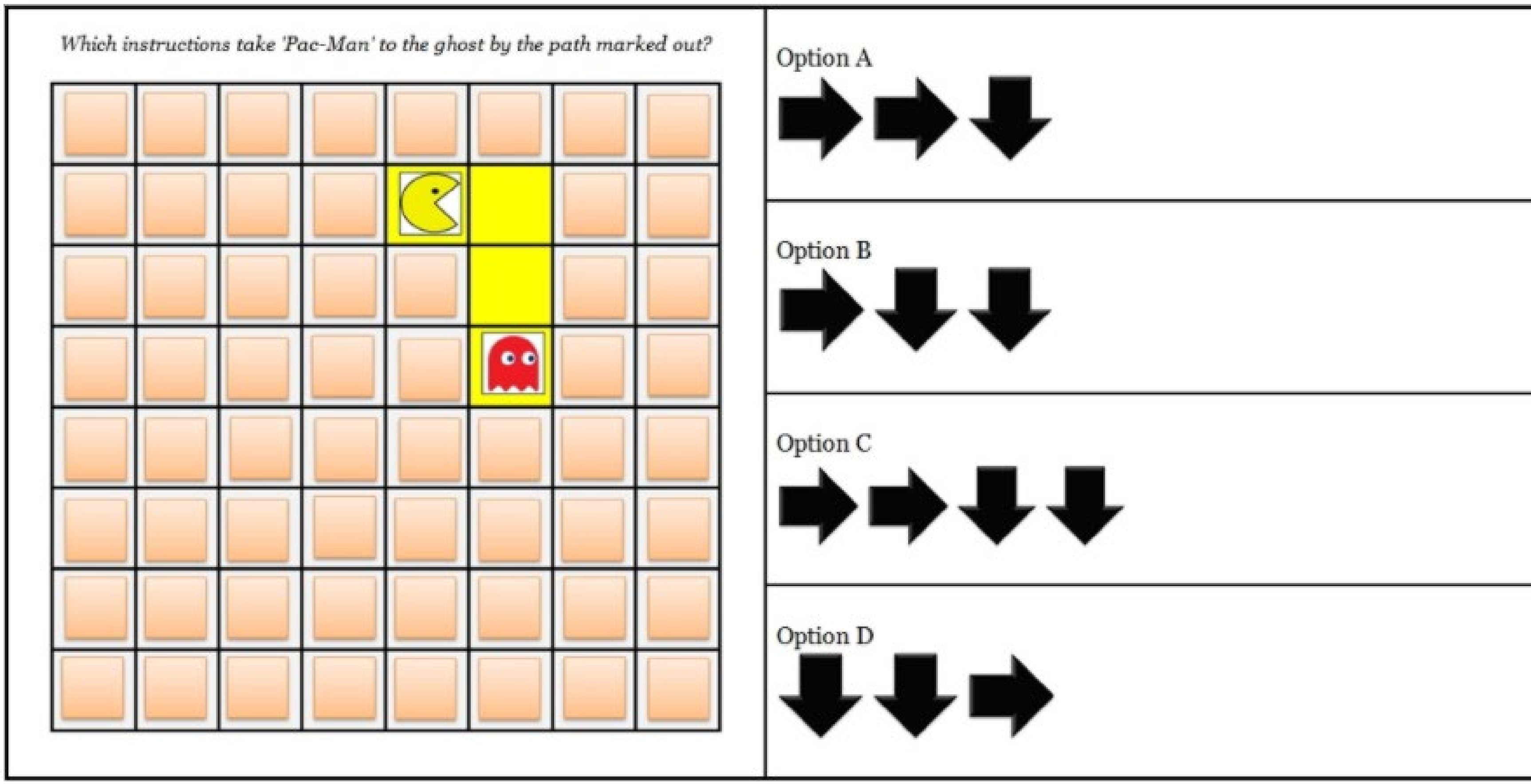
20

Most Used Measure of Computational Thinking

- *Computational Thinking Test (CTt)* (Román-González, 2015)
  - 28 items administered online
  - Each item presented in a “maze” or a “canvas” interface; addresses one or more computational concepts, ordered in increasing difficulty:
    - Basic directions and sequences/Loops/If/If else/While/Simple functions
  - Test has strong psychometrics (reliability and validity)
  - Appropriate for grades 6-10

Example Item, Computational Thinking Test (CTt)

Which instructions take Pac-Man to the ghost by the path marked out?



Existing Validated Measures of Computational Thinking

Preschool	***TechCheck*** (Relkin et al., 2020) ▪ Unplugged assessment of computational thinking that does not require knowledge of coding
Elementary school	***Computational Thinking Scale*** (Korkmaz et al., 2017) ▪ 29 items covering creativity, cooperativity, algorithmic-critical thinking, problem-solving
Middle school	***Computational Thinking Abilities − Middle Grades Assessment*** (Wiebe et al., 2019) ▪ Combines items from CTt and Bebras (described in later slide)
	Computational Thinking Performance Test (Mindetbay et al., 2019) ▪ 50 multiple-choice items covering logical thinking, generalization, abstraction
High school	***Computational Thinking in STE***M (Weintrop, 2014) ▪ Measures STEM students' computational thinking skills, highlighting the power of computation in the practice of scientific and mathematical inquiry
	Computational Thinking Tool (Yagci, 2018) ▪ Multiple-choice test measuring problem-solving, creative thinking, algorithmic thinking, cooperative learning, and critical thinking
	Computational Thinking Self-Efficacy Survey (Weese & Feldhausen, 2017) ▪ Student perception of their ability to think computationally

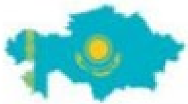
Example Assessment of CT Skills

- *Computational Thinking Performance Test* (Mindetbay et al., 2019)
  - 50 multiple-choice questions and a Computational Thinking Scale questionnaire
  - Covers logical thinking, generalization, and abstraction
  - Shown to be valid and reliable

Computational Thinking Performance Test: Example Item

Appendix A

Sample question on Pattern figures



VIVA
FIFA
LIFE
FIVE

Each letter is represented by a certain figure.
Identify the letters that are represented by the combination

A *VLIAF*

B *FLIAL*

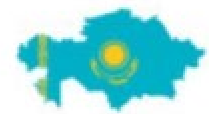
C *VLIFA*

D *VLIAV*

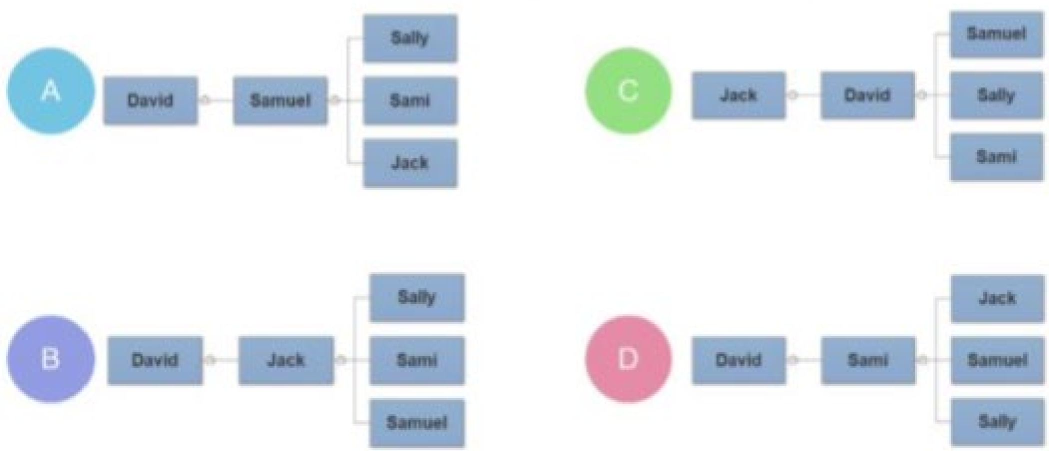
Computational Thinking Performance Test: Example Item

Appendix C

Sample question on Abstraction



David is granddad to Sally.
Sally's brother is Sami.
Jack's father is David. Identifythe correct diagram



Example of Survey Measure of Student CT Self-Efficacy

- *Computational Thinking Self-Efficacy Survey* (Weese & Feldhausen, 2017)
  - Student perception of their ability to think computationally
  - Covers problem-solving, computer programming skills, computer programming practices, and computer programming impact

Computational Thinking Self-Efficacy Survey: Example Item

When solving a problem I...			I can write a computer program which...		
1	create a list of steps to solve it	Algorithms	10	runs a step-by-step sequence of commands	Algorithms
2	use math	Algorithms	11	does math operations like addition and subtraction	Algorithms
3	try to simplify the problem by ignoring details that are not needed (3)	Abstraction	12	uses loops to repeat commands	Control Flow
4	look for patterns in the problem	Abstraction	13	responds to events like pressing a key on the keyboard	Control Flow
5	break the problem into smaller parts	Problem Decomposition	14	only runs commands when a specific condition is met	Control Flow
6	work with others to solve different parts of the problem at the same time	Parallelization	15	does more than one thing at the same time	Parallelization
7	look how information can be collected, stored, and analyzed to help solve the problem	Data	16	uses messages to talk with different parts of the program	Parallelization
8	create a solution where steps can be repeated (8)	Control Flow	17	can store, update, and retrieve values	Data
9	create a solution where some steps are done only in certain situations (9)	Control Flow	18	uses custom blocks	Abstraction

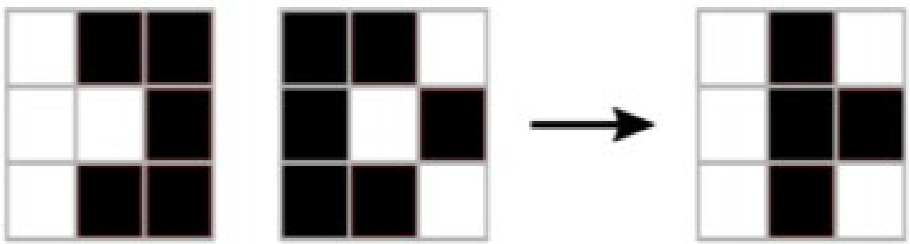
Source of New Measures: *Bebras Challenge*

- *Bebras Challenge*
  - *Bebras* is an international initiative aiming to promote computer science (computing) and computational thinking among school students grades 1-12
  - *Bebras* challenges consist of a set of short problems called “Bebras tasks” and are delivered online
  - A challenge has two types of tasks:
    - Multiple-choice questions
    - Interactive problems

https://www.bebras.org/?q=goodtask

Bebras Challenge: Example Item

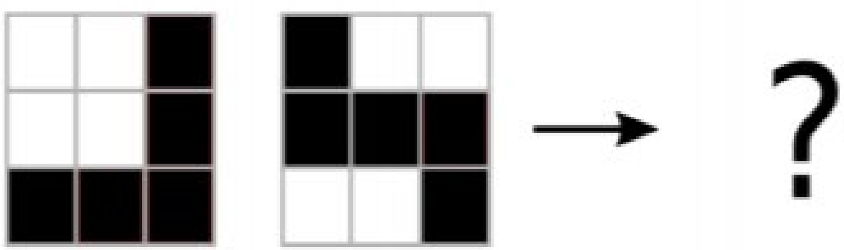
Combining Card A and Card B, you get Card C:



Card A Card B Card C

Question:

How many black cells will Card F have after combining Card D and Card E?



Most Used Measure of CS Proficiency

- *Advanced Placement Computer Science Principles Exam*
  - Exam assesses student understanding of the computational thinking practices and learning objectives outlined in the AP –CS Principles course framework.
- Exam includes the Create performance task (program coding) and the End of Course multiple choice exam
  - The Create performance task requires at least 12 hours of dedicated class time for students to complete. The end-f-course exam is 2 hours long and includes 70 multiple-choice questions.

Existing Validated Measures of CS Proficiency

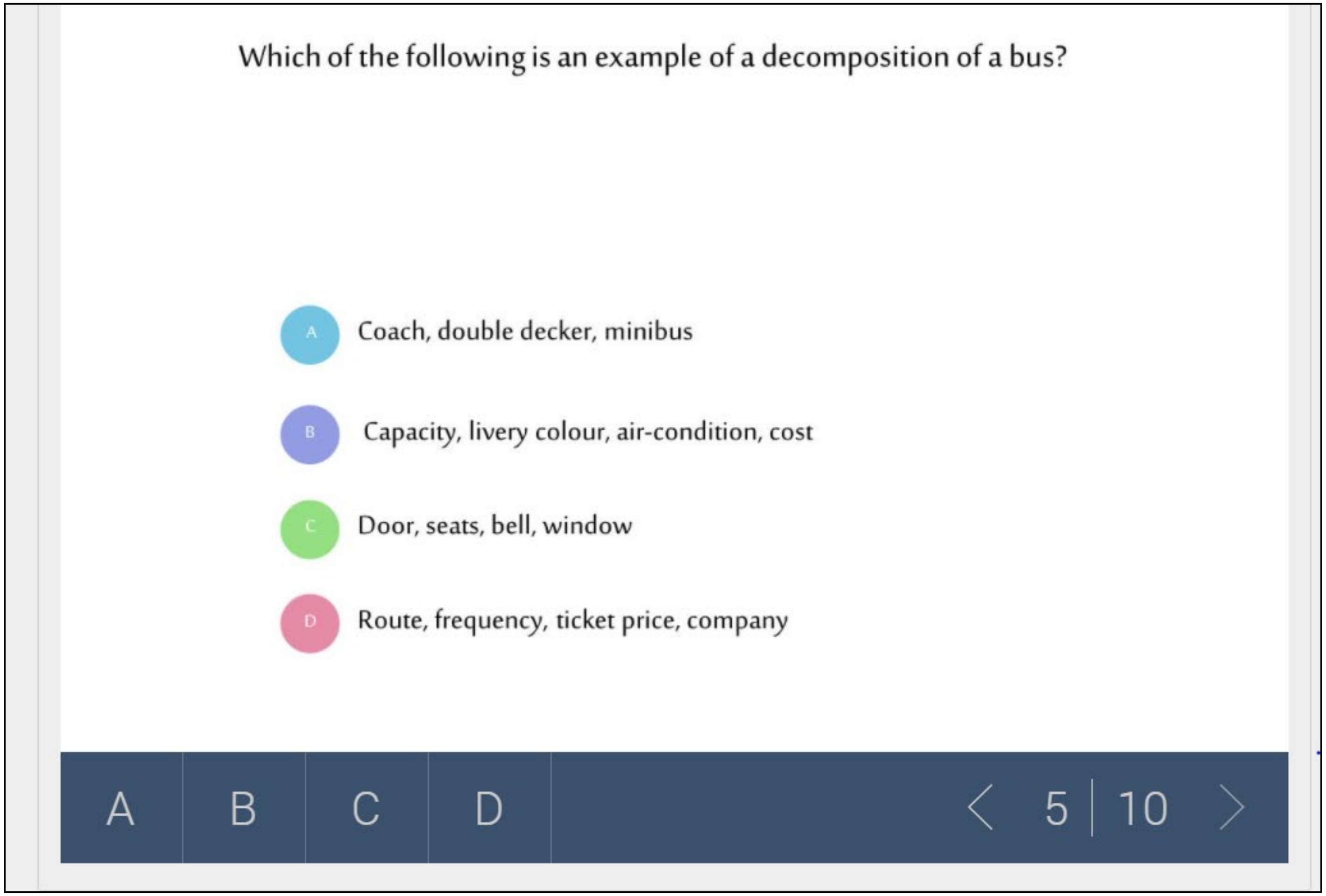
Preschool	None
Elementary school	*Dr. Scratch* (Moreno-León et al., 2015). Analytical tool that evaluates *Scratch* projects in a variety of computational areas (Abstraction, parallelization, logical thinking, synchronization, flow control, user interactivity, and data representation). [*Scratch* is a block-based visual programming language and website targeting primarily children to help them learn code by creating online projects using a block-like interface.]
	Bebras Challenge tasks Project Challenge tasks
Middle school	Bebras Challenge tasks Project Challenge tasks
High school	AP-Computer Science Principles Exam ((ETS)
	Bebras Challenge tasks Project Challenge tasks

Project Quantum: Item Pool for New Measure Development

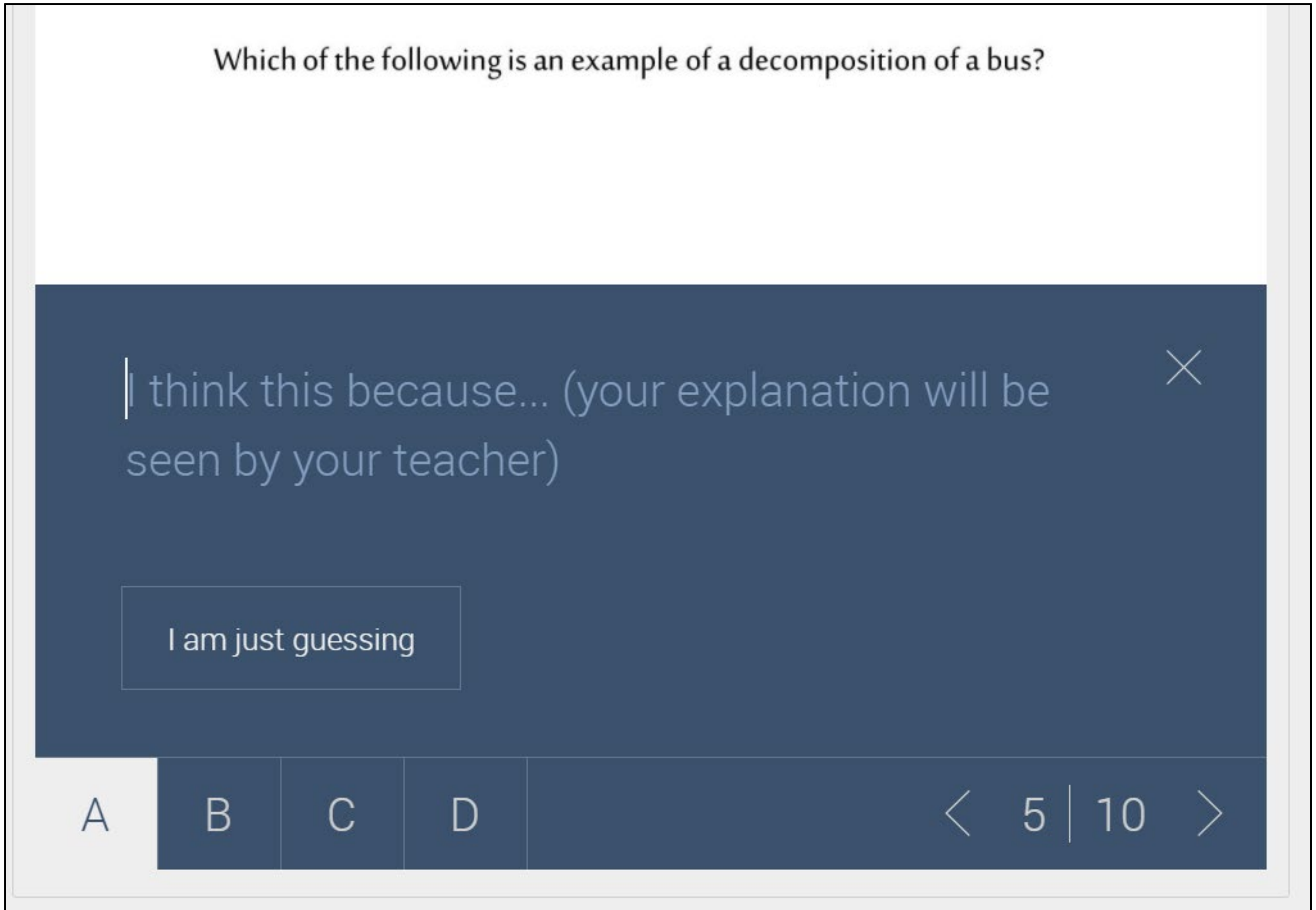
- Collection of computing quizzes
- Crowd sourcing a bank of high-quality multiple-choice questions for assessing computing in schools
- Items in three elements of computing:
  - Computer science (foundations)
  - Information technology (applications)
  - Digital literacy (implications)
- For every question in the item bank, data are provided on how many students have answered this question, how many chose each answer (correct and incorrect), and what explanations they gave

https://diagnosticquestions.com/Quantum

Project Quantum CS: Example Item



Project Quantum CS: Example Item



Student selects "A" and explains why: *"Because they are all types of buses."*

Project Quantum CS: Example Summary Test Score

Student sees # correct out of 10 items

Project Quantum CS: Example Student Explanations

Students Who Chose Correct Response	Students Who Chose "A"
I think this because decomposition means taking apart and doors and seatbelts are all things in a bus	Because if we brake it into chunks, these are the parts of a bus
I think this because you are breaking things into smaller chunks	Because those word are the word they normaly call a bus
I think this because they are in all different places around the bus not next together and I think that decomposition means breaking something down or moving the position so that all the things in this answer are in different places	I think this because I think decomposition means breaking big problems into smaller problems so a double-decker bus go into a coach and a coach into a minibus
I think this because there is a door inside of the bus, chairs are also a luxury inside of a bus. Also you have to use a bell to mark an area or destination you want to get off at	I think this because those are different kinds of buses broken down

Gaps/Challenges in CS/CT Measure Development

- Computer science proficiency
  - Challenge of measuring coding skills in cost-efficient way
- Computational thinking and computer science proficiency
  - Need for measures that can be used with neuro-diverse students
  - Need to validate measures on subgroups of special focus (e.g., English language learners)

Student and Teacher Attitudes and Beliefs About STEM, CS, and CT

39

Role of Mediators in Achieving Student Learning

- Mediators capture changes in student and teacher attitudes/beliefs that precede improved CS/CT skills
  - How important are these mediators in the theory of change?
  - Do these changes make it more likely that students will learn new skills?
- Potential Mediators
  - Student interest/engagement in STEM/CS
  - Student and teacher sense of self-efficacy
  - Student belief in importance of STEM/CS for future career and college options

State of Student Measures

- Large number of measures on different facets of students' attitudes and beliefs
  - Interest in / Intentions to learn STEM/CS
  - Career interest in STEM/CS
  - Sense of self-efficacy about CS, CT, coding

Existing Surveys on Student Attitudes/Beliefs Towards STEM

Survey Measure	Grade Level		
	K-5	6-8	9-12
Upper Elementary School and Middle/High School Student Attitudes Toward STEM (S-STEM) Surveys (Faber et al., 2013)		✓	✓
Adapted Middle School Students' Attitudes to Mathematics, Science, and Engineering Survey (Hirsch et al., 2007)		✓	
Math and Science Engagement Scales (Wang et al., 2016)		✓	✓
STEM Career Interest Survey (C-SIS) (Kier et al., 2013)		✓	
Student Attitudes Towards STEM – Upper Elementary (Friday Institute, 2012)	✓		
Student Attitudes Towards STEM – Middle/High School (Friday Institute, 2012)		✓	✓
Student Attitudes Toward Science, Technology, Engineering, and Math (S-STEM) (Unfried et al., 2015)	✓	✓	✓

Existing Surveys on Student Attitudes/Beliefs Towards Computer Science

Survey Measure	Grade Level		
	K-5	6-8	9-12
Computer Attitude Measure for Young Students (CAMYS) (Asil et al., 2008)	✓		
Elementary Student Coding Attitudes Survey (Mason & Rich, 2019)	✓		
Attitudes Towards Computing Scale (Wanzer et al., 2019)			✓
Computer Science Interest Survey (Blouin, 2011)			✓
Attitudes About Computers and Computer Science (Drobnis, 2010)			✓
Student Computer Science Attitude Survey: CS Principles (Haynie, 2017)			✓
Interest, Confidence, and Intentions to Learn Computing (Weston et al., 2019)			✓

State of Teacher CS Measures

- Not a highly developed field in CS education
- Research in other areas of STEM (esp. science) point to the definition, relevance, and significance of measurement targets such as:
  - Content knowledge/Subject matter knowledge (Arzi & White, 2004; Gess-Newsome & Lederman, 1995)
  - Pedagogical knowledge (Grossman, 1990)
  - Pedagogical content knowledge (Schulman, 1986)
  - Attitudes, beliefs, and dispositions (Jones & Carter in 2007 Res Sci Ed Handbook)

Theoretical Framework for Teacher CS Measures

Attitudes, Beliefs, Dispositions

- CS education represents a novel undertaking for many teachers, raising the importance of attitudes, beliefs, and dispositions as mediators to reform
- Decades of findings in science (esp. K-5) have shown avoidance (Harlen, 1997), low self-efficacy (Jones & Levin, 1994), and epistemological issues (Lederman, 1992), with a generalist population teaching a less familiar, less accountable content area at scale

- Robust theoretical frameworks and applications point to worthy measurement targets in the area of teacher attitudes, beliefs, and dispositions
  - Social Identity Theory (Gee, 2000-2001; Avraamidou, 2016 edited volume)
  - Self-Efficacy (Bandura, 1997; Deehan, 2017 review of STEBI findings)
  - Social Cognitive Career Theory (Lent, Brown, & Hackett, 1994; Roller et al., 2020 SCCT STEM instrument)

Existing Measures of Teacher CS Attitudes and Beliefs

Attitudes, Beliefs, Dispositions

- *Teacher Efficacy and Attitudes Toward STEM (T-STEM) Survey* (Friday Institute, 2012)
  - Confidence and self-efficacy in STEM subject content and teaching, use of technology in the classroom, 21st century learning skills, leadership attitudes, and STEM career awareness
- *Teacher Beliefs about Coding and Computational Thinking (TBaCCT) Scale* (Rich, Larsen, & Mason, 2020)
  - Teacher self-efficacy for coding, computing, and computational thinking

Existing Measures of Teacher CS Pedagogical Knowledge

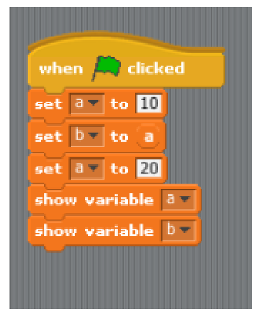
- *Computer Science Pedagogical Content Knowledge Instrument* (Yadav & Bergs, 2019)

PCK

1 PEDAGOGICAL CONTENT KNOWLEDGE INSTRUMENT

PCK1:

Phil has worked on the following Scratch example and when you ask him about the value ‘b’, he suggested that b would be 20.



How would you help Phil understand this code segment?

	I would...	Definitely Will	Probably Will	Probably Will Not	Definitely Will Not
PCK1.1:	...explain to Phil what happens in the code.	○	○	○	○
PCK1.5:	...ask Phil questions until he recognizes his error.	○	○	○	○
PCK1.6:	...explain to Phil how and when variables change.	○	○	○	○
PCK1.7:	...ask Phil about his intentions when writing the code.	○	○	○	○

Leveraging EIR to Advance the STEM/CS/CT Field

48

Disseminating New Measures

- CSEdResearch.org hosts instruments, for sharing with proper attribution
- EIR projects are developing potentially promising surveys and protocols
- Researchers and evaluators in the broader educational community (particularly in CS) can benefit from your work
- Submit by emailing monica@csedresearch.org or visiting https://www.csedresearch.org/submit-to-repository/

csedresearch.org

Questions

Resources

***Computer Science and Computational Thinking. EIR Small Group Workshops*. Abt Associates. July 2020**

Will be uploaded on same site as conference webinar slides.

BOLD THINKERS DRIVING REAL-WORLD IMPACT

www.abtassociates.com

Contact

EIR Evaluation TA Team

EIREvalTA@AbtAssoc.com

Abt Associates
BOLD THINKERS DRIVING REAL-WORLD IMPACT

www.abtassociates.com