Principled Assessment of Computational Thinking

Measuring CT Concepts, Practices and Perspectives Through Evidence-Centered Assessment

CTE 2017, Hong Kong

Eric Snow, Ph.D., Senior Researcher
Linda Shear, Deputy Director
Center for Technology in Learning
Propositions

• Computational thinking curricula are growing quickly
  => Stakeholders want to know if they’re effective

• Assessments have not yet kept pace
  => And they’re hard to develop

• Principled Assessment Design can help
Principled Assessment Design: Developing a chain of evidence

What students do

Claims about students’ knowledge and skills

Design pattern
Overview

Computational Thinking
Measuring Computational Thinking Outcomes
Assessment Arguments & Evidence-Centered Design (ECD)
ECD Applications
  \textit{PACT}
  \textit{CoolThink@JC}
Closing Comments
Computational Thinking
Wing (2006, 2011) defined computational thinking as the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by a computer.
Computational Thinking

Grover & Pea (2013) identified these features of CT in their extensive literature review:

- abstractions and pattern generalizations
- systematic processing of information
- symbol systems and representations
- debugging and systematic error detection
Computational Thinking

Workforce studies (e.g., Malyn-Smith & Lee, 2012) indicate CT involves **collaborating** and **engaging in a creative process** to:

- solve problems
- design products
- automate systems
- improve understanding by defining, modeling, qualifying and refining systems
Brennan and Resnick’s (2012) framework divides CT into three interrelated areas:

- CT Concepts >>> CS content knowledge
- CT Practices >>> logical thinking/problem solving skills
- CT Perspectives >>> interest, motivation, self-efficacy
Computational Thinking Practices

“Ways of being and doing” that students should demonstrate when learning and exhibiting computer science knowledge, skills, and attitudes

- Algorithmic thinking
- Reusing & remixing
- Testing & debugging
- Abstracting & modularizing
Measuring CT Outcomes
Impacts of CT Curricula?

Despite the proliferation of CT-focused programs, there is still a lack of strong empirical studies that examine the impact of curricula on students’ CT learning outcomes.

• Lye and Koh (2014): only 27 studies examining computational thinking skills as outcomes

• Challenge for teachers: locating valid and reliable assessments of computational thinking for their classrooms
Assessment Arguments & Evidence-Centered Design
Assessment Questions

What knowledge, skills, or other attributes should be assessed?

>>> What do we want students to know and be able to do?

What behaviors or performances should reveal those attributes?

>>> What counts as evidence?

What tasks or situations should elicit those behaviors?

>>> How do we get students to produce that evidence?
Evidence-Centered Assessment Design

ECD is a framework for assessment design and development.

Views assessment as a process of gathering evidence to support an argument about what a student knows and can do.

Provides a structure for an approach that incorporates validity evidence into the assessment design process.

Particularly useful when the knowledge/skills to be measured involve complex, multistep performances, such as those required in computational thinking.
Students interact with tasks, performances evaluated, feedback created.

- What is important about this domain?
- What work and situations are central in this domain?
- What KRs are central to this domain?

How do we represent key aspects of the domain in terms of assessment argument.

Design structures: Student, evidence, and task models.

Manufacturing “nuts & bolts”: authoring tasks, automated scoring details, statistical models.

Students interact with tasks, performances evaluated, feedback created.

---

From Mislevy & Riconscente, 2006
Applications of Evidence-Centered Design
Principled Assessment of Computational Thinking

Exploring Computer Science
Principled Assessment of Computational Thinking (PACT)

Create design templates for supporting CT assessment development for secondary CS.

>>> Computational Thinking Design Patterns

Design, develop, validate assessments of computational thinking practices aligned with the ECS curriculum.
What’s in a design pattern?

<table>
<thead>
<tr>
<th>Focal Knowledge, Skills, and other Attributes (FKSAs)</th>
<th>Specific knowledge and skills that we want to make inferences about</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional KSAs</td>
<td>Other knowledge and skills that may be required for successful performance, but aren’t what we’re trying to measure</td>
</tr>
<tr>
<td>Potential observations</td>
<td>Features of the things students say, do, or make that constitute evidence of their performance</td>
</tr>
<tr>
<td>Potential work products</td>
<td>Artifacts of the student’s performance we can see</td>
</tr>
<tr>
<td>Characteristic features</td>
<td>Aspects of assessment situations that will evoke the desired evidence or are required to support the task</td>
</tr>
<tr>
<td>Variable features</td>
<td>Aspects of assessment situations that can be varied in order to shift difficulty or emphasis</td>
</tr>
</tbody>
</table>
### Focal Knowledge, Skills, and other Attributes (FKSAs)

- Ability to state a problem in order to identify its inputs and outputs
- Ability to decompose a problem into multiple subproblems, and specify how solving one subproblem will help generate a solution to the problem as a whole
- Ability to create a computational artifact given a purpose or intent
- ...

### Additional KSAs

- Knowledge of a specific programming language
**Design and implement creative solutions and artifacts**

| Potential observations | • Degree to which the solution is related to the identified purpose  
| | • Degree to which the solution addresses the problem  
| | • Level of complexity of the solution  
| | • … |
| Potential work products | • Description of the design process  
| | • Computational solution  
| | • Description or explanation of the solution  
| | • Description of the debugging process  
| | • Trace of the debugging process  
| | • … |
Computational Thinking Design Patterns

- Analyze the effects of developments in computing
- Design and implement creative computational solutions and artifacts
- Design and apply abstractions and models
- Analyze computational work (both own and others)
- Communicate computational thought processes, procedures and results to others
- Collaborate with peers on computing activities
CoolThink@JC

Bringing computational thinking to Hong Kong students in Primary 4-6.

Target outcomes based on Brennan and Resnick (2012) CT Framework:

>>> CT Concepts, CT Practices, CT Perspectives

SRI used ECD to develop student assessments and a survey that are aligned with these outcomes.
Specified focal knowledge and skills aligned with CT Practices

<table>
<thead>
<tr>
<th>CT Practices</th>
<th>Focal Knowledge &amp; Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithmic Thinking:</td>
<td>1. Ability to state a problem in terms of what is given and what the end result should be</td>
</tr>
<tr>
<td>Articulating a problem solution in well-defined rules and steps</td>
<td>2. Ability to recognize and define computational problems</td>
</tr>
<tr>
<td></td>
<td>3. Ability to create a specification for the implementation of a solution</td>
</tr>
<tr>
<td></td>
<td>4. Etc.</td>
</tr>
</tbody>
</table>
Darren wants to create a program to draw the following picture:

![Picture](image)

Darren uses the following steps to make the program. He finds that some things are not in the correct place.

- Draw Grass
- Move to the bottom left
- Draw Dog
- Move to the bottom right
- Draw Ball
- Move to the top left
- Draw Cat
- Move to the bottom middle
- Draw Tree
- Move to the top right
- Draw Flowers
a. Put a check mark in front of the things that are not in the correct place.

- Dog
- Ball
- Cat
- Tree
- Flowers

b. Put the steps in order so that they will put all of the things in the correct place. You can move the steps by dragging each block into the space provided.
Closing Comments
Closing Comments

Growing consensus on facets of computational thinking.

Validated measures of CT outcomes are needed to provide stakeholders with high-quality evidence of student learning.

ECD is particularly relevant when the knowledge/skills to be measured involve complex, multistep performances, such as those required in computational thinking.
Links for more information

ECD: https://ecd.sri.com/

PACT: https://pact.sri.com/

Design patterns: http://bit.ly/2u6t0Nw

ECS assessments and rubrics: https://pact.sri.com/ecs-assessments.html
Thank you!