Assessing Computational Thinking

Eric Snow
Center for Technology in Learning
SRI International

Assessment Breakout Session
CE21 Community Meeting
January 6-8th, 2014
Orlando, FL
Overview

• Computational Thinking
• Computational Thinking Assessments
• Lessons on the Road to Validity
• Discussion
Principled Assessment of Computational Thinking is applying the Evidence Centered Design (ECD) approach to create assessments that support valid inferences about computational thinking practices.
Computational Thinking

Emerging consensus on characteristics of CT, but emphasis, specification and scope vary across use contexts.
Computational Thinking

Various proposed definitions suggest related constructs:

• algorithmic thinking
• modeling
• symbolic representations
• working with patterns
Computational Thinking

Workforce studies (e.g., Malyn-Smith & Lee, 2012) indicate CT involves:

• solving problems
• designing products
• automating systems
• defining, modeling, qualifying and refining systems, processes or mechanisms generally through the use of computers
At the K-12 level, computational thinking often emphasizes problem solving and data representations (e.g., CSTA/ISTE, 2011).
Computational Thinking Practices

New high school curricula (e.g., CS Principles, ECS) emphasize “computational thinking practices”.

This reflects an orientation toward not just an internal, individual “thinking” but “ways of being and doing” that students should demonstrate when learning and exhibiting computer science knowledge, skills, and attitudes.
## Computational Thinking Practices

<table>
<thead>
<tr>
<th>Example CS Concepts</th>
<th>Inquiry Skills</th>
<th>Communication &amp; Collaboration Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td>Evaluate</td>
<td>Publish</td>
</tr>
<tr>
<td>Programming</td>
<td>Explore</td>
<td>Present</td>
</tr>
<tr>
<td>Recursion</td>
<td>Investigate</td>
<td>Build Consensus</td>
</tr>
<tr>
<td>Abstraction</td>
<td>Explain</td>
<td>Discuss</td>
</tr>
<tr>
<td>Debugging / Testing</td>
<td>Elaborate</td>
<td>Distribute Work</td>
</tr>
<tr>
<td>Variables</td>
<td>Modeling</td>
<td>Lead/Manage Teams</td>
</tr>
</tbody>
</table>
Inquiry

Skills

CS Concepts

Inquiry Skills
# Computational Thinking Practices

<table>
<thead>
<tr>
<th>Example CS Concepts</th>
<th>Example Inquiry Skills</th>
<th>Communication &amp; Collaboration Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td>Evaluate</td>
<td>Publish</td>
</tr>
<tr>
<td>Programming</td>
<td>Explore</td>
<td>Present</td>
</tr>
<tr>
<td>Recursion</td>
<td>Analyze</td>
<td>Build Consensus</td>
</tr>
<tr>
<td>Abstraction</td>
<td>Explain</td>
<td>Discuss</td>
</tr>
<tr>
<td>Debugging / Testing</td>
<td>Elaborate</td>
<td>Distribute Work</td>
</tr>
<tr>
<td>Variables</td>
<td>Model</td>
<td>Lead/Manage Teams</td>
</tr>
</tbody>
</table>
Computational Thinking Practices

Inquiry Skills

CS Concepts

Communication & Collaboration Skills

Computational Thinking Practices
# Computational Thinking Practices

<table>
<thead>
<tr>
<th>Example CS Concepts</th>
<th>Example Inquiry Skills</th>
<th>Example Communication &amp; Collaboration Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td>Evaluate</td>
<td>Publish</td>
</tr>
<tr>
<td>Programming</td>
<td>Explore</td>
<td>Present</td>
</tr>
<tr>
<td>Recursion</td>
<td>Analyze</td>
<td>Build Consensus</td>
</tr>
<tr>
<td>Abstraction</td>
<td>Explain</td>
<td>Discuss</td>
</tr>
<tr>
<td>Debugging / Testing</td>
<td>Elaborate</td>
<td>Distribute Work</td>
</tr>
<tr>
<td>Variables</td>
<td>Model</td>
<td>Lead/Manage Teams</td>
</tr>
</tbody>
</table>

Integration
## Computational Thinking Practices

<table>
<thead>
<tr>
<th>Exploring Computer Science</th>
<th>Computer Science Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyze the effects of developments in computing</td>
<td>Analyzing Problems and Artifacts</td>
</tr>
<tr>
<td>Design and implement creative solutions and artifacts</td>
<td>Developing Computational Artifacts</td>
</tr>
<tr>
<td>Apply abstractions and models</td>
<td>Abstracting</td>
</tr>
<tr>
<td>Analyze their computational work and the work of others</td>
<td>Analyzing Problems and Artifacts</td>
</tr>
<tr>
<td>Connect computation with other disciplines</td>
<td>Connecting Computing</td>
</tr>
<tr>
<td>Communicate computational thought processes, procedures and results to others</td>
<td>Communicating</td>
</tr>
<tr>
<td>Collaborate with peers on computing activities</td>
<td>Collaborating</td>
</tr>
</tbody>
</table>
Computational Thinking Practices

Noncognitive Skills?

Interpersonal

Intrapersonal

Computational Thinking Practices
## Noncognitive Skills

<table>
<thead>
<tr>
<th>Interpersonal Skills</th>
<th>Intrapersonal Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>Self-efficacy</td>
</tr>
<tr>
<td>Teamwork/collaboration</td>
<td>Self-concept</td>
</tr>
<tr>
<td>Leadership</td>
<td>Persistence</td>
</tr>
<tr>
<td>Cultural awareness</td>
<td>Organization</td>
</tr>
<tr>
<td>Tolerance for diversity</td>
<td>Time management</td>
</tr>
</tbody>
</table>
Computational Thinking Practices

Cognitive Skills

Noncognitive Skills
Computational Thinking

The Common Core State Standards include standards related to computational thinking practices in mathematics such as problem and abstraction.

The Next Generation Science Standards include standards dealing with engineering design and describe “using mathematical and computational thinking” as an essential practices for modeling and analyzing and interpreting data.
Computational Thinking Assessments

Postsecondary Education
- FCS₁
- CS Major Field Assessment
- GRE Subject Assessment

Secondary Education
- AP CS
- CS Principles
- Exploring Computer Science
Computational Thinking Assessments

Challenges

- Programming language
- Conceptual vs. syntactic knowledge
- Cognitive and noncognitive factors
- CT in non-CS domains (e.g., science, mathematics)
- Limited validity evidence to support desired uses
Lessons on the Road to Validity

Assessment Validity

“...degree to which evidence and theory support the interpretations of test scores [in the context of proposed test uses].”

*Standards for Educational and Psychological Testing*, pg. 9 – 1999: AERA, APA, NCME
Lessons on the Road to Validity

Lesson #1:

Assessments are *not* “plug and play”
Lessons on the Road to Validity

Lesson #1:
Assessments are *not* “plug and play”

Need to *check compatibility between instruction and assessment* in terms of the targeted knowledge and skills, what counts as evidence of those targets, and the types of tasks that elicit the evidence.
Lessons on the Road to Validity

Lesson #2:

Validity is use-specific
Validity is Use-Specific

Students → Assessment → Scores → Decisions

KSAs → Evidence → Tasks → Design

Assessment Scores Decisions
Lesson #3:

Assessment design focuses on evidence, not just creating innovative tasks.
Lessons on the Road to Validity

Lesson #3:
Assessment design focuses on evidence

Assessment Design

• What **KSAs** do I want to assess?
• What would be **evidence** of those KSAs?
• What **tasks** would elicit the correct evidence?
Final Comments

• The train has already left the station.

• Future opportunities & challenges:
  – Putting the cart before the horse, particularly with regard to validity
  – Connecting implementation with learning outcomes
  – Relating cognitive and noncognitive factors
  – Game-based assessment & learning analytics
Discussion

• Possible discussion questions:
  – How are you assessing CT in your settings?
    • Targeted knowledge and skills?
    • Types of evidence?
    • Tasks to elicit evidence?
  – What are some of the successes/challenges you have experienced when assessing CT in your settings?
  – Promising new approaches for assessing CT?