Understanding Computational Thinking

Principled Assessment of Computational Thinking

Data Sources

- Next Generation Science Standards
- NCPR Framework Document
- CSTA Standards
- CS Principles Big Ideas
- Exploring CS
- Expert Panels

Future Work

Sample NGSS Engineering Performance Expectations Compared with Computational Thinking Practices

Performance Expectations (NGSS)

- HS - ETS1-1: Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants
- HS - ETD1-2: Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

Focal KSA

- Analyze the Effects of Developments in Computing
- Communication Thought Processes and Results to Others
- Design and Apply Abstractions and Models

Computational Thinking Practice

- Students have to understand problems to which computing can be applied in order to define problems and solve them responsibly. Inspiration for solutions can come in part from understanding applications in other domains.
- In order to verify the solutions meet important needs, communication and collaboration are critical.
- Representations not only help communicate the criteria and constraints, but also to help a designer think through them.

ECS Unit | Focal KSA
--- | ---
Unit 1: Human Computer Interaction | Students are able to explain why an object is or is not a computer
Unit 2: Problem Solving | Students are able to compare the tradeoffs between different algorithms for solving the same problem
Unit 3: Web Design | Students are able to apply abstraction to separate style from content during web page design and development
Unit 4: Introduction to Programming | Students are able to evaluate debugging and testing methods in terms of how they relate to the problem or program

We can Re-interpret the Framework’s Science & Engineering Practices as CT Practices

1. Asking questions (for science) and defining problems (for engineering)
2. Constructing explanations (for science) and designing solutions (for engineering)
3. Analyzing the tradeoffs between different algorithms for solving the same problem
4. Applying abstractions and models
5. Analyzing and interpreting data
6. Designing and implementing Creative Solutions and Artifacts

Notes
- Reusing code involves such exploration. “Data” can include data produced by computation. Computing emphasizes testing, debugging, and critical thinking about the limits of computing.
- We have to understand problems to which computing can be applied in order to define problems and solve them responsibly.
- We have to be able to see how problems in other domains can be solved computationally.