Introducing Computer Science to the Public

Six Computational Thinking Practices

In addition to the Seven Big Ideas underpinning Computer Science, the commission identified six computational thinking practices -- activities that computer scientists engage in.

Contents

Tell me and I forget. Show me and I remember. Involve me and I understand.
-- Chinese Proverb

The Six Computational Thinking Practices typify the kinds of activities computer scientists engage in, and by extension, must typify the learning outcomes of the computer science course. These are companions to the Seven Big Ideas.

1. Analyzing the Effects of Computation

- Identification of existing and potential innovations enabled by computational technology.
- Identification of ethical implications of developments in computing.
- Identification of the impacts (positive and negative) of computing innovations on society.
- Analysis of implications of design decisions.
- Evaluation of the usability of a computational artifact.
- Characterization of connections between human needs and computational functionality.
- Explanation of relevant intellectual property issues.

2. Creating Computational Artifacts

- Creation of an artifact chosen by the student as relevant and interesting.
- Design of a solution to a stated problem.
- Selection of an appropriate approach to solve a problem.
- Appropriate use of pre-defined algorithms.
- Appropriate use of programming constructs and data structures.
- Evaluation of an artifact using multiple criteria.
- Location and correction of errors.
- Use of appropriate technique to develop a computational artifact.

3. Using Abstractions and Models

- Explanation of how data, information and knowledge are represented for computational use.
- Use of simulation to investigate posed/existing questions and develop new questions.
- Selection of algorithmic principles at an appropriate level of abstraction.
- Use of different levels of abstraction.
- Specification of the design for a model/simulation.
- Use of data abstractions.
- Collection or generation of data appropriate to a phenomenon being modeled.
- Comparison of generated data to an empirical sample.

4. Analyzing Problems and Artifacts

- Identification of problems and artifacts that have a given property.
- Comparison of tools available to solve a problem.
- Evaluation of a proposed solution to a problem and implications of that solution's use.
- Analysis of solution tradeoffs with appropriate justification of possible solutions.
- Analysis of the result of a program.
- Evaluation of characteristics of problems and artifacts.

5. Communicating Processes and Results

- Explanation of the meaning of results.
- Description of the impact of a technology or artifact.
- Summarization of the behavior of a computational artifact.
- Explanation of the design of an artifact.
- Description of technology or artifact.
- Justification of the appropriateness and correctness.

6. ## Working Effectively In Teams

- Application of effective teamwork practices.
- Collaboration of participants.
- Production of artifacts that depend on active contribution from multiple participants.
- Documentation describing the use, functionality, and implementation of an artifact.

Contact: snyder at cs dot washington dot edu