Computational Thinking in Junior High School

Soledad A. Ulep
University of the Philippines
National Institute for Science and Mathematics Education Development

Some Background Information

- Computer applications in Technology Livelihood Education in the curriculum
- Robotics Club in a regional science high schools

Possible integration of Computational Thinking in the Mathematics Curriculum

“Computational thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer – human or machine – can effectively carry it out.” (Jeannette Wing, computer scientist)

The goal of the K to 12 Mathematics Curriculum is the development of learners’ problem solving and critical thinking skills.

Elements of Computational Thinking*

- Computational thinking concepts include:
  - logic and logical thinking
  - algorithms and algorithmic thinking
  - patterns and pattern recognition
  - abstraction and generalization
  - evaluation and automation

*Computational Thinking: A Competency whose Time has Come by Shuchi Grover and Roy Pea, 2017.

Elements of Computational Thinking

- Computational thinking practices include:
  - problem decomposition
  - creating computational artifacts
  - testing and debugging
  - iterative refinement (incremental development)
  - evaluation and automation
  - collaboration and creativity

*Computational Thinking: A Competency whose Time has Come by Shuchi Grover and Roy Pea, 2017.

Examples: Computational Thinking Concepts

- Logic and logical thinking : use of if, and, or
- Algorithms and algorithmic thinking: flowcharting
- Pattern and pattern recognition: determining patterns in statistical data

Examples: Computational Thinking Concepts

- Abstraction and generalization: algebraic equations

- Evaluation: correctness and appropriateness of solution, satisfying the constraints/conditions, efficiency

- Automation: determining which part of the solution is better solved by machines

Examples: Computational Thinking Practices

- Problem decomposition: solving a multi-step problem
- Creating computational artifacts: simulation
- Testing and debugging: evaluating if solution works
- Incremental development (iterative refinement): frequent testing and debugging
- Collaboration and creativity: thinking out of the box and finding alternative solutions to problems

“Programming is especially useful platform for teaching computational thinking since it brings together several of the elements – concepts and practices that are central to CT.”

Thank you.