Olympiads in Informatics, 2017, Vol. 11, Special Issue, 65–70

DOI: 10.15388/ioi.2017.special.10

Computational Thinking: A 21st Century Skill

Yahya TABESH
Sharif University of Technology, Iran
e-mail: tabesh@sharif.ir

In memory of ***Seymour Papert*** *(1928-2016)*

1. Prologue

Computational thinking is a 21st century skill that is becoming ever-more important in today's increasingly technological world. It should appear formally in the K-12 curriculum, but the inherent vagueness of computational thinking necessitates a long journey to the classroom: contents must be developed under a new pedagogical means and be tested before gradually being incorporated into the curriculum. Training of teachers to adopt the new pedagogical means is also a prerequisite to incorporation. Thus, a faster route to help students could be considering computational thinking as an extracurricular activity, gradually developing and elaborating it, and finally imposing it in the curriculum once a solid framework has been achieved.

In this paper, we introduce the pillars of computational thinking and present practical suggestions, namely computational mathematics and artificial intelligence labs, as platforms for developing computational thinking in K-12. The outlined approach is conceptual and high level – it should not be considered an action plan for the education system, but rather an ideological proposal in need of more elaboration for execution.

2. Computational Thinking Playground

Computational Thinking can be defined as the thought processes involved in formulating a problem and expressing its solution in such a way that a computer can effectively carry out. It is a way of solving problems, designing systems, and understanding human behavior that draws on concepts fundamental to computer science. To flourish in today's world, computational thinking has to be a fundamental part of the way people think and understand the world. (Kamvar, 2015). It is essential for school children and should be part of the K-12 curriculum, but we should first consider its roots and the pedagogical model that can be a basis for content development.

The essence of computational thinking is what we can do while interacting with computers, as extensions of our mind, to create and discover (Knuth, 1980). Such a concept was envisioned by Seymour Papert in his book Mindstorms (Papert, 1980) as follows (Fig. 1).

S. Papert focused on two aspects of computation: first, how to use computation to create new knowledge, and second, how to use computers to enhance thinking and change patterns of access to knowledge. More recently, J. M. Wing brought a modified approach and new attention to computational thinking (Wing, 2006) (Fig. 2).

J. M. Wing considers computational thinking as a fundamental skill for everyone's analytical ability along with reading, writing and arithmetic. Wing's paper was welcomed by the community in all levels, especially in K-12, which was highly responsive and began development of applications addressing teenagers, many of which are on the way. We will visit the fundamental ideas of S. Papert but will go to problem solving as a means of instilling CT.

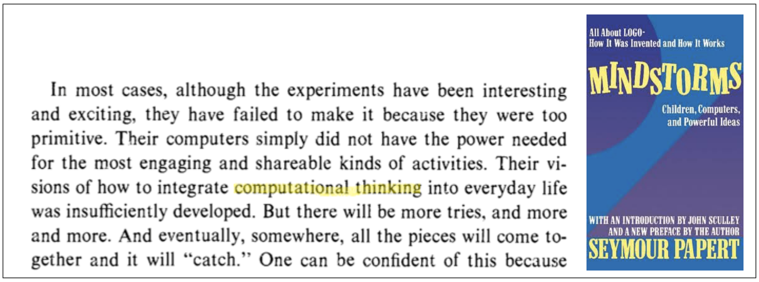
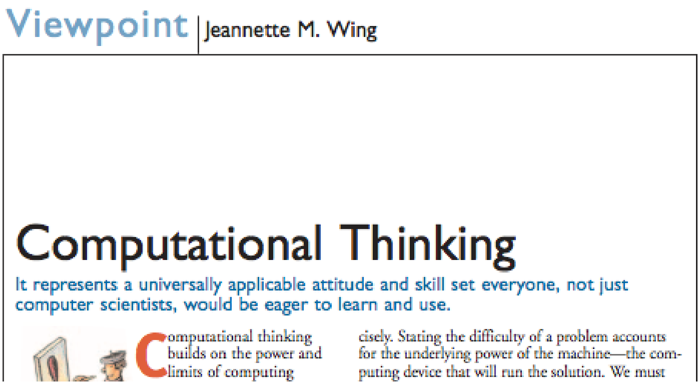

In most cases, although the experiments have been interesting and exciting, they have failed to make it because they were too primitive. Their computers simply did not have the power needed for the most engaging and shareable kinds of activities. Their visions of how to integrate computational thinking into everyday life was insufficiently developed. But there will be more tries, and more and more. And eventually, somewhere, all the pieces will come together and it will "catch." One can be confident of this because

Fig. 1. Computational thinking is a legacy of Seymour Papert reflected in Mindstorms (1984).


Viewpoint | Jeannette M. Wing

Computational Thinking

It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.

Computational thinking builds on the power and limits of computing

cisely. Stating the difficulty of a problem accounts for the underlying power of the machine—the computing device that will run the solution. We must

Fig. 2. J.M. Wing's paper appeared in the Communication of the ACM in 2006.

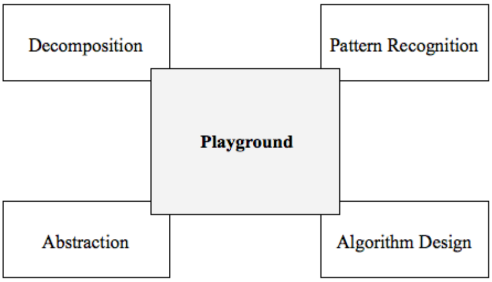


Fig. 3. "Playground" enriches the 4-stages.

S. Papert connected computational thinking and digital pedagogy to the modern approach in education initiated by Jean Piaget. J. Piaget was a developmental psychologist best known for pioneering the learning theory known as constructivism; in brief, he says that learners construct new knowledge in their minds, from the interaction of their experiences with previous knowledge. S. Papert developed the theory of constructivism, adding the notion that the learning is enhanced when the learner is engaged in "constructing a meaningful product."

We consider computational thinking based on S. Papert's enhancement of thinking, specifically with a problem solving approach. Put simply, computational thinking combines critical thinking with computing power as the foundation for innovating solutions to real-life problems.

Computational thinking involves a four-stage problem-solving process as follows:

- **Decomposition**: Analyzing the problem to break it up into smaller parts.
- **Pattern Recognition**: Observing patterns, trends, and regularities in data.
- **Abstraction**: Identifying the underlying principles that generate perceived patterns.
- **Algorithm Design**: Developing step-by-step instructions for solving the problem.

We have enriched and connected the stages with a "playground" as a place for experimental problem solving (Fig. 3). In this model, the playground is an easily accessible place where learners can experiment numerically, geometrically and procedurally by modeling or backtracking in simple or similar cases, all while looking for flows, patterns, symmetry, parity, invariants, recursion, etc. The playground fosters an environment that promotes cognitive learning over force-feeding.

3. Computational Thinking in School

Now that we have established the sense and significance of computational thinking, we must look at how to help these ways of thinking take shape for teenagers in middle and high school. We should consider opportunities to help students after-school as an extracurricular activity, but we will see how computational thinking may go to the classroom as well (Lee, 2011).

We present some brief ideas to apply as hands-on labs in an extracurricular approach for middle and high school students. Labs are drawn in two domains: computational mathematics and artificial intelligence. Across these subjects we will consider computational thinking as the backbone of problem solving.

3.1. *Computational Mathematics Lab*

Computational mathematics lab is a learning environment focused on skilled mathematics such as discrete math, data science and algorithm design, using computational thinking processes for problem solving. It is a place for experimental learning through workshops and projects, and subsequent reflection in showcases and mathematics festivals. In computational mathematics lab, learners are not in a one-path trail; the environment is designed to foster richer and deeper interactions than the ones commonly seen in schools. It is a learning environment utilizing teamwork and collaboration, where learners create ideas in the form of collaborative projects and interact genuinely, since their mathematical work belongs to them and to real life. Much of the fun lies in sharing projects, allowing for modification and experimentation across projects and teams.

Workshops are the main pathway for empowerment in the lab; they are organized by computational mathematics skills, and learners are partners and active agents in the learning process in a teamwork approach. Contents are presented in a hands-on and step by step problem solving fashion through the four-step computational thinking process.

Projects bring opportunity for creativity enhancement. Learners should formulate a problem as a project in a team and use computational thinking methodologies to tackle it thoroughly. Results of projects will be presented in festivals and showcases, which not only foster further interaction between teens but also attract new teens to join the lab.

3.2. *Artificial Intelligence Lab*

With immense computing power available, artificial intelligence (AI) has become a hot field and is now reflected in different domains such as robotics, Internet of Things (IoT) etc. In the new approach, gathering data through smart sensors, analyzing the data to make meaningful inferences, and applying these inferences to control systems in a decision-making process are essential. Robotics and IoT are means of applying computational thinking processes to formulate problems and find solutions. AI lab is an empowering, high-tech environment for teenagers, where they can be trained in new skills in a heuristic and hands-on fashion, and, through projects, develop their own creativity.

Basic components of robotics and sensors of IoT applications should be provided in the lab. Materials for constructing new objects are also necessary, and a 3D printer is essential. Furthermore, a cloud system to gather data and software tools to analyze said data also should be provided. Upon arrival to the lab, teens should have access to robotics and IoT components and in a heuristic approach should learn about their functionality and applications. Teens should also be trained in teams with the cloud system and data analysis tools. Teamwork in the training process will also bring the opportunity for them to learn from one another. In the second phase, also in teams, they should formulate a problem, and by applying computational thinking process, they should look for a solution, which they can construct and test by gathering data on the cloud and applying data analysis tools.

Again, presenting the outcomes of projects in tech festivals and showcases and being recognized for their explorations are good incentives for teens to join the AI lab.

4. Epilogue

Integrating computational thinking into K-12 settings has been attracting greater attention lately, despite the lack of a formal description of what "learning to think computationally" actually looks like among today's youth. The youth can engage in key aspects of computational thinking within a rich learning environment, using computational mathematics and artificial intelligent labs as means to solve problems and create original products.

Certain frameworks and procedures should be developed to introduce CT in K-12, and systematic assessment procedures are still needed to describe the developmental progression of computational thinking through computational mathematics and AI labs. However, the two labs outlined should provide a solid base for further teen exploration in computation.

As a foundation moving forward, the "use-modify-create" framework offers a helpful progression for developing CT over time. It illustrates the benefits arising from engaging youth with progressively more complex tasks and giving them increasing ownership of their learning.

References

Kamvar, S. (2015). *Syntax & Sage*. Farmer & Farmer Press.

Knuth, D.E. (1980). Algorithms in Modern Mathematics and Computer Science. *Stanford Department of Computer Science Report No. STAN-CS-80-786*.

Lee, I. *et al.* (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32–37.

Papert, S. (1980). *MINDSTORMS, Children, Computers, and Powerful Ideas*. Basic Books.

Wing, J.M. (2006). Computational Thinking. *Communications of the ACM*, 49(3) 33–35.

Y. Tabesh. Distinguished faculty at the Sharif University of Technology. Co-founder of the Iranian National Informatics Olympiad. Co-founder of the Iranian Mathematical Olympiad. Erdös award winner 2010.