CAMBRIDGE Mathematics

IN PARTNERSHIP WITH arm School Program

ESPRESSO DOPPIO

ISSUE 19 APRIL 2019

RESEARCH, FILTERED BY CAMBRIDGE MATHEMATICS

TALKING POINT:

WHAT DOES RESEARCH SUGGEST ABOUT APPROACHES AND TOOLS THAT PROMOTE COMPUTATIONAL THINKING IN THE CLASSROOM?

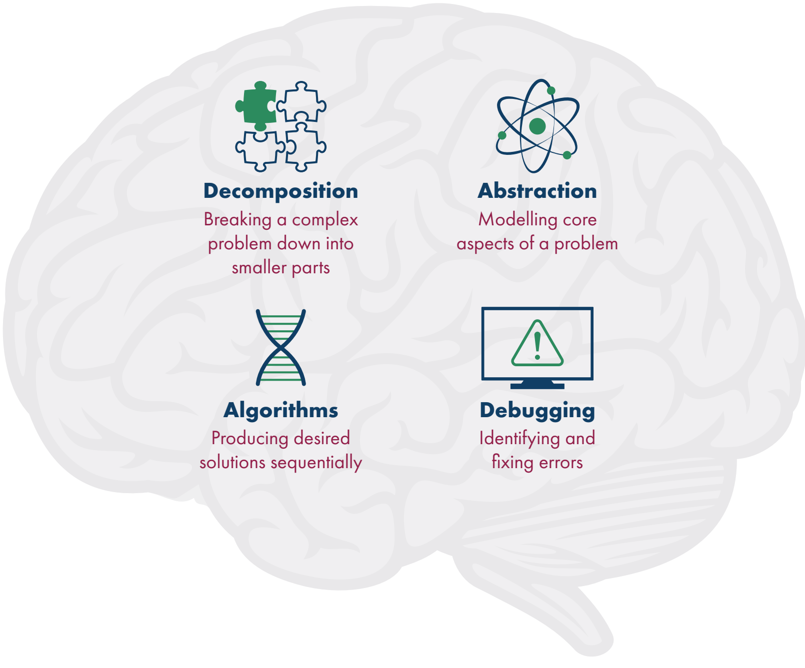
IN SUMMARY

- CT is a fundamental skill useful for diverse subjects and can be promoted through various approaches and tools
- Further research is needed to investigate how these approaches and tools can be used effectively in the classroom to support the learning and teaching of CT
- Educational robotics can be used for learners as young as 4 to learn basic CT and programming concepts. This approach has been shown to enhance mathematics and science understanding that is beyond the development of CT skills.
- It is important to consider learners' developmental stages when using robotics tools to promote CT
- Visual programming languages are good introductory tools to deliver basic computer science concepts
- When using visual programming languages, teachers can encourage learners to share thought processes so that the focus is on thinking processes as well as output
- 'Unplugged' activities are alternative economical ways to promote CT without the use of a computer

Computational thinking

'Not simply to think like a computer'

(Wing, 2006)



Adapted from Shute, Sun & Asbell-Clarke (2017)

1

Computational thinking (CT) uses cognitive processes that are useful across school subjects and also in the real world, where problems are often ill-defined[3]. 'The thoughtful use of computational tools and skillsets can deepen learning of mathematics and science content'[4] and some researchers advocate teaching it as a fundamental skill applicable across most subjects[1]. There are several approaches and tools—each with different implications—to foster CT. However, further research is needed to investigate how they can be used effectively in the classroom to support learning and teaching.

IMPLICATIONS: CT uses thought processes that are helpful for problem-solving across school subjects and in the real world

Further research is needed to shed light on how these approaches and tools can promote CT effectively in the classroom

> 'Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability'
>
> **Wing, 2006**

> '...computational thinking includes a broad range of mental tools and concepts from computer science that help people solve problems, design systems, [and] understand human behavior'
>
> **National Research Council, 2011**

CAMBRIDGE UNIVERSITY PRESS · UNIVERSITY OF CAMBRIDGE Faculty of Mathematics · UNIVERSITY OF CAMBRIDGE Faculty of Education · Cambridge Assessment

2

Educational robotics offers engaging experiences at all levels of education[7] and even to learners as young as 4-6 years old[5]. As 'objects to think with'[6], robotic tools create an exploratory environment for learners to design, problem-solve and program while constructing knowledge. This enhances mathematics and science understanding while contributing to their STEM aspirations[7]. Learners who participated in a robotics curriculum demonstrated significant gains in CT knowledge and skills[8]. However, it is important for teachers to utilise developmentally appropriate robotic tools to promote computational concepts, practices and perspectives successfully[7].

IMPLICATIONS: Educational robotics can be a meaningful way to introduce simple CT and programming concepts to learners as early as 4-6 years old

Robotics activities provide a platform for learners to learn-by-doing, which enhances learners' motivation

Using robotics can help enhance understanding in mathematics and science while contributing to STEM aspirations for learners

The success of promoting CT through robotic tools may depend on whether they are compatible with the developmental stages of learners

3

Visual programming languages such as *Scratch* and *Alice* are gaining popularity as tools to promote CT in the classroom. Due to their strong pedagogical link to *constructionism*[6], these tools encourage learners to create artefacts (i.e. graphics and animation) and thereby the development of CT concepts, practices and perspectives. Research suggests that child-friendly programming languages such as these are typically easy enough for beginners to pick up computer science concepts (*low-floor*) while allowing them to create sophisticated programs (*high-ceiling*)[9]. In the process, learners explore CT concepts through experimentation or tinkering. Higher-attaining learners usually tinker more, suggesting that this approach might be more appropriate for them[10] whilst more scaffolding may be needed with lower-attaining learners. To avoid hindering conceptual understanding, learners should be encouraged to explain their thought processes in their own language before coding. This ensures that thinking processes are equally valued as outcomes[11 12].

IMPLICATIONS: Visual programming languages are good introductory tools for learners to practice CT skills while learning basic computer science concepts

Whilst it is beneficial to higher-attaining learners to use problems that facilitate tinkering, it is important to provide more scaffolding for lower-attaining learners

Teachers should ask learners to explain their thought processes explicitly prior to block-based coding so that thinking skills are both valued and fostered

4

The approaches above emphasise the hardware and programming aspect of CT; 'unplugged' activities provide alternative ways to develop learners' problem-solving skills in CT[13]. For example, using playful board (or paper-based) games, competitions, and 'magic' tricks demonstrate CT processes[14] (see csunplugged.org). This approach may be particularly useful for schools that do not have other resources or infrastructure in place (i.e. computers, technological devices, trained computing teachers). CT concepts delivered by unplugged activities can be embedded in diverse subject areas, or used towards transitioning to plugged-in activities[15]. Teachers and students consider CT unplugged lessons as a good alternative to regular, on-line computer science class[16] and research suggests that learners who took part in unplugged activities improved their CT skills significantly compared with learners who did not[17].

IMPLICATIONS: 'Unplugged' activities such as engaging board games are alternative ways to deliver CT concepts without the requirement of a computer

They can be an economical way to enhance CT skills, which could be particularly useful for schools that have fewer resources

Teachers and students find unplugged activities a motivating way to learn CT—this motivation extends to increased interest and confidence in learning computer science and mathematics

Rina P.Y. Lai, 2019

REFERENCES

1. Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.
2. National Research Council. (2011). *Report of a workshop on the pedagogical aspects of computational thinking*. Washington, DC: The National Academies Press.
3. Cuny, J., Snyder, L., & Wing, J. (2010). Demystifying computational thinking for non computer scientists. Unpublished Manuscript.
4. Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127–147.
5. Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education, 72*, 145–157.
6. Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books, Inc.
7. Penmetcha, M. R. (2012). *Exploring the effectiveness of robotics as a vehicle for computational thinking* (Master's thesis). Retrieved from ProQuest Dissertations & Theses A&I (Order No. 1529729).
8. Witherspoon, E. B., Higashi, R. M., Schunn, C. D., Baehr, E. C., & Shoop, R. (2017). Developing computational thinking through a virtual robotics programming curriculum. *ACM Transactions on Computing Education (TOCE), 18*(1),1-20.
9. Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behaviour, 41*, 51-61.
10. Rose, S., Habgood, J., & Jay, T. (2017). An exploration of the role of visual programming tools in the development of young children's computational thinking. *Electronic Journal of E-Learning, 15*(4), 297–309.
11. Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717–3725.
12. Lai, R.P.Y. (in press). Exploring the cognitive mechanism of computational thinking and its educational implications. In *The Proceedings of International Conference on Computational Thinking Education 2019 Conference*. Hong Kong: The Education University of Hong Kong.
13. Looi, C.-K., How, M.-L., Longkai, W., Seow, P., & Liu, L. (2018). Analysis of linkages between an unplugged activity and the development of computational thinking. *Computer Science Education, 28*(3), 255–279.
14. Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: school students doing real computing without computers. *New Zealand Journal of Applied Computing and Information Technology, 13*(1), 20–29.
15. Caldwell, H., & Smith, N. (2016). *Teaching computing unplugged in primary schools: Exploring primary computing through practical activities away from the computer*. Learning Matters.
16. Faber, H. H., Wierdsma, M. D. M., Doornbos, R. P., Ven, J. S. van der, & Vette, K. de. (2017). Teaching computational thinking to primary school students via unplugged programming lessons. *Journal of the European Teacher Education Network, 12*,13–24.
17. Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, 65–72.

CAMBRIDGE UNIVERSITY PRESS

UNIVERSITY OF CAMBRIDGE Faculty of Mathematics

UNIVERSITY OF CAMBRIDGE Faculty of Education

Cambridge Assessment