Using Classroom Practice as "an Object to Think with" to Develop Preservice Teachers' Understandings of Computational Thinking

Deirdre Butler(✉) and Margaret Leahy

Dublin City University, Institute of Education, Dublin, Ireland
deirdre.butler@dcu.ie

Abstract. Irrespective of what approach is taken to the development of computational thinking, or at what age it is introduced, the teacher is central to ensuring that the children they work with develop computational thinking. It is therefore essential that their teachers are adequately prepared to include computational thinking as part of their pedagogical classroom practices. Moreover, it is argued that this preparation should begin at pre-service level. Adopting a constructionist perspective of learning, this paper presents and discusses the findings from research that investigated preservice teachers' understandings of computational thinking, having completed a specialism in digital learning, the final activity of which entailed using computational tools with children in the classroom as part of a primary school science curriculum. Findings indicate that working with the children in the classroom helped the preservice teacher develop their own understandings of what computational thinking (CT) looks like "in action", enabling them to reflect more deeply on the fundamentals of CT and on how to use CT in their own classroom practice as qualified teachers.

Keywords: Pre-service teachers · Computational thinking · Constructionism

1 Introduction

Computation and the development of computational thinking at primary school level is an area of research that is still in its infancy [1]. In Ireland, as elsewhere, current debates centre on whether there should be a computer science curriculum at primary level with an explicit focus on computational thinking [1, 2]. Does computational thinking become a battle cry for coding in primary education [3]? Or should young people be able to develop and use computational thinking concepts in their problem-solving activities as part of subject areas other than computer science, e.g. [4]? Since 2018, the national curriculum body in Ireland, the National Council for Curriculum and Assessment (NCCA), has begun to investigate how computational thinking can be introduced. Irrespective of what approach is taken to the development of computational thinking, or at what age it is introduced, the teacher is central to ensuring that the children they work with develop computational thinking. If young people are to develop computational thinking, it is

Published by Springer Nature Switzerland AG 2020
T. Brinda et al. (Eds.): OCCE 2020, IFIP AICT 595, pp. 56–65, 2020.
https://doi.org/10.1007/978-3-030-59847-1_6

essential that their teachers are adequately prepared to include computational thinking as part of their pedagogical classroom practices [5, 6]. Moreover, it is argued that this preparation should begin at preservice level [7] so that preservice teachers not only develop understandings of computational thinking but are also introduced to ways they can design learning opportunities for their students to develop computational thinking [8].

The study reported in this paper contributes to this debate. Adopting a constructionist perspective of learning, it presents and discusses the findings from ongoing research, specifically focusing on one group of final year Bachelor of Education (B.Ed.) students who have completed a major specialism in digital learning. The aim of the research was to investigate preservice teachers' understandings of computational thinking having completed their specialism in digital learning, the final activity of which entailed using computational tools with children in the classroom as part of the primary school science curriculum [9].

2 Review of Underpinning Literature

Underpinned by constructionist learning theory and computational thinking perspectives, the primary focus of the major specialism in digital learning is the development of preservice teachers' understanding of the theoretical and practical concepts of computational thinking and coding.

2.1 Constructionism

An extension of constructivism, constructionism is both a theory of learning and a strategy for education [10]. Originating in the work of Seymour Papert in the 1980s, constructionism shares constructivism's connotation of learning as building knowledge structures but goes beyond constructivism by emphasising that learning is facilitated by constructing tangible artefacts or objects, which can then be shared and discussed with others [11]. As such, constructionism sees learners as active builders of their own knowledge and asserts that people learn with particular effectiveness when they are engaged in constructing personally meaningful artefacts, "whether it's a sand castle on the beach or a theory of the universe" [11, p. 1]. These artefacts which Papert [12] describes as "objects to think with" (p. 12) support the development of concrete ways of thinking and learning about phenomena. The ability to manipulate these objects, to repeatedly make adjustments and refinements or experiment with them to see how they work lends itself to a concrete style of reasoning. This, Papert argues, changes the process of learning to one which is iterative and cumulative, embracing both planning and bricolage styles. Turkle and Papert [13] refer to this 'validity of multiple ways of knowing and thinking' as "an epistemological pluralism" (p. 129).

Constructionism also draws attention to the social nature of learning, noting that activities such as making, building or programming through which the learner produces artefacts that others can see and critique provide a rich context for learning. The artefacts are a means by which others can become involved in the thinking process while, at the same time, the learner's thinking benefits from multiple views and discussions [14].

In this way, the artefacts or 'objects to think with' provide a link between sensory and abstract knowledge, and between the individual and the social worlds. Shared knowledge is constructed when artefacts and shared understanding are coupled through cycles of representing and interpreting [15]. Through engaging in conversation around their own or another's artefact, the development of a shared understanding is enabled and the foundation for new understandings is provided [16]. Constructionism thus implies a *process* of building, both in the sense of building artefacts and building new understandings.

2.2 Computational Thinking

The concept of computational thinking also originates in the work of Papert [12], when he introduced the "idea of the computer being the children's machine that would allow them to develop procedural thinking through programming" [17, p. 2]. However, it was not until 2006 and the publication of Wing's [18] seminal article that the concept of computational thinking came to prominence. Describing computational thinking as a "fundamental skill" for everyone, Wing defined it as the thought process of formulating and solving problems by "drawing on the concepts fundamental to computer science" (p. 33) when "equipped with computing devices" (p. 35). Within this broader context, she outlined the central components of computational thinking, including algorithms, abstraction, decomposition and automation, all of which can be found in many contexts and disciplines and which assist learners in developing problem-solving skills. Researchers have continued to put forward a number of definitions of computational thinking as they built on the work of Wing but have failed to agree an accepted definition of computational thinking [e.g. 1, 4, 7, 17, 19–21]. However, it is broadly accepted that computational thinking is a thought process that utilises the elements of abstraction, generalisation, decomposition, algorithmic thinking and debugging, i.e. detection and correction of errors [1] (see Table 1).

Table 1. Core elements of computational thinking.

Element	Definition
Abstraction	Reducing unnecessary details, highlighting the relevant details to make the process simpler and easier to understand
Algorithmic thinking	Devising a step by step solution to a problem
Decomposition	Breaking down complex problems into manageable smaller problems
Generalisation	Looking for a general approach to a class of problems
Debugging	Skill to identify, remove, and fix errors

A range of dispositions or attitudes have been identified which some have claimed are integral to the development of computational thinking. Brennan and Resnick [22], for example, refer to these dispositions as computational practices and computational perspectives. Computational practices are the "problem solving practices that occur in the process of programming" (p. 53) and include: iterative and incremental, testing and debugging, reusing and remixing, and abstracting and modularising. Computational perspectives relate to the "student's understandings of themselves, their relationships to others, and the technological world around them" (p. 53).

A number of implementation frameworks have also been put forward. While most of these frameworks focus on post-primary and third level, a small number have been presented for primary level [e.g. 1, 22, 23]. Across these frameworks, one of the most frequent methods of providing the opportunity to engage in computational thinking in primary classrooms is through the use of programming languages such as Scratch [e.g. 22]. Although scholars advocate the introduction of computational thinking early at primary school level [24], to date, little attention has been accorded to preservice primary school teachers. While there have been some examples of planned structured teacher preparation programmes [25–27], and it is widely accepted that the development of computational thinking for preservice K-8 teachers should be integrated with pedagogical content knowledge [27], framework or models that focus explicitly on preservice teachers are not yet available [28].

2.3 Computational Thinking in a Constructionist Learning Environment

From a constructionist perspective, computational thinking can be thought about in much the same way as Papert viewed computer programming; that is, computational thinking is both a skill to learn and a way to learn – "to create, discover, and make sense of the world, with digital technologies as extensions and reflections of our minds" [29].

Computational tools can be a powerful medium for creating contexts for constructing knowledge. However, in keeping with Papert's idea of engaging with "powerful representations", what is essential to consider when designing a learning environment is not so much what programming language and/or computational materials to use, but what personally meaningful ideas the programming language and materials can enable the children to develop and how those ideas will develop computational thinking and form new ideas about the subject area (e.g. mathematics, science). Activities and learning situations should be developmentally appropriate for the children and grounded in meaningful contexts [14].

Drawing together the ideas presented in the literature review, the authors ensured in their design of the major specialism to provide immersive learning experiences for the students which were underpinned by constructionist principles. The students were introduced to a range of expressive computational materials (e.g. Scratch and Lego WeDo) through which the students were scaffolded to progressively develop understandings of computational thinking. As module facilitators, the authors continuously worked alongside the students using a range of pedagogical strategies and practices to help them construct and reflect on their emerging understandings of computational thinking while linking these to classroom practice.

3 Research Design

3.1 The Context and Participants

The study was conducted with 18 preservice teachers who had taken a major specialism in digital learning as part of their Bachelor of Education (B.Ed.) programme at the Institute of Education, Dublin City University (DCU IoE). The aims of the major specialism were to: expose students to the concepts of CT, raise awareness of the definitions; enable them to build their own interpretation based on experience, literature reading, class discussions, and hands-on learning experiences using a range of computational materials; and finally, to make learning concrete and relevant through the completion of assignments that related to their future teaching and the primary school curriculum. Modules were accordingly designed to include a range of computational materials and contexts to explore strategies that support interest-driven, project-based, collaborative approaches to learning. The final module, 'Designing & Learning with Digital Technologies', was designed to enable students to translate their learning into practice [30]. To this end, students, in groups of three, were required to design a unit of work comprising three workshops of two hours' duration in which they introduced coding and computational thinking concepts using Lego robotic materials within the context of the Primary School Science Curriculum [9] to the 4th or 5th class (children aged 10-11 years) in local schools. The final module was also designed to provide the students with an opportunity to engage in research, thus providing "the foundation of their practitioner-based enquiry stance in the future" [30, p. 23]. As part of classwork, they had been introduced and/or revised participant observation as a data collection method, the deductive approach to data analysis and the ethics of classroom-based research (students had previously completed a research methods course as part of their B.Ed. programme). Students documented the classroom-based work as part of the module assessment

3.2 Methodology

The aim of the study was to consider from the perspective of a preservice teacher how their understandings of computational thinking had developed as a result of completing a major specialism in digital learning. The study adopted a qualitative approach. Data collection methods comprised document analysis and group interviews:

a) Each of the 18 students wrote reports of their classroom-based experience as the module assignment. Students gave the authors permission to use these reports as part of the data corpus.
b) Group interviews with student teachers took place at the end of the semester. Each group interview, approximately 20 min with four to five students, was carried out by the authors. The aim of the interviews was to probe the students' experiences, understandings and reflections in relation to computational thinking.

A typological analysis framework [31] was used as a starting point for analysis. The main themes from the literature review in relation to computational thinking were used to generate typologies, and initial data processing took place within these categories. The categories were re-examined after coding to ensure that they were justified by the data, or if the data excluded contained insights contrary to that proposed. Overall, decisions were driven by the data and, where necessary, new categories of adjustment added. This iterative process helped reduce the data to a small set of themes that then lent themselves to the final narrative.

4 Findings

Based on the analysis of the data, what was immediately evident was the preservice students' use of CT terminology, demonstrating their development of a "computational thinking language" [32]. They were able to define computational thinking and elaborate on computational processes and practices using classroom examples from their experiences and observations. They stressed how important it was that they had sufficient time and opportunity to engage in problem-based tasks using the computational materials to develop their understandings of CT, before they worked with the primary school children. This first-hand experience helped them understand challenges children could have and they were able to plan for and adapt their classroom practices accordingly. They also observed how computational thinking and, in particular, logical reasoning developed as the children completed the learning tasks. This included some of the practices of computational thinking such as experimentation and iteration, testing and debugging as well as the development of skills or perspectives such as collaboration, communication (expression) and critical thinking (questioning) [22]. As stated in one focus group:

> …it brought it to life I think, for me anyway… even when we were revising at the beginning just the different components of the computational thinking and being like oh yeah, that's what this is. But then going into the classroom and seeing, like linking the theory of it with children actually putting it into practice really got me to understand the definition of it more. (Focus Group 1)

In her essay, Student#4 noted and tracked the development of logical reasoning across the three weeks, stating at the outset that "there was variance in the children's ability to think logically. The majority of children employed basic strategies to solve problems that arose when building and programming their models". However, by the end of the project, she documented the children using strategies of logical reasoning such as predicting, analysing, creating and correcting their algorithm and the build of their model:

> In the final week, the children were challenged to make their frogs move faster adapting both their builds and code. One child responded with … *"Well, if you want to get a car to go faster you need more speed… the speed comes from the acceleration so…. we could try giving him (robot) more acceleration and he might go faster."* This child analysed the problem linking it with their own knowledge and experiences "to predict the behavior of…. programs" [33, p. 9]. The child broke down the problem and thought logically about the elements that affect the speed of a car and transferred this knowledge to their own code. By saying, "might", this insinuated that this solution may not work and the child would have to think of an alternative solution, this implies that the child would be tinkering with the problem. (Assignment_Student#4)

The preservice teachers observed that the children's self-confidence grew and although they were given challenging tasks, they persevered. They observed that a contributing factor to this perseverance was the design of the learning task (inquiry-based science focus/constructionist principles) combined with the choice of computational materials. Together, these facilitated the creation of a collaborative learning environment as the materials enabled children to externalise their thinking and allowed others to see a physical representation of their understanding. It thus enabled the children to give feedback to each other based on the model they had constructed and the program they had developed. The preservice teachers remarked that they began to realise the importance of using computational materials that enabled children to work with others so that they could benefit from the knowledge and insights that other people brought to the situation. As student teachers said:

> The findings suggest that clear design-based learning activities need to be planned to ensure that children can experience all the computational thinking skills needed when coding and building these projects. Lego Wedo was a significant factor in this project as it enabled the children to work in groups designing a project and coming up with a solution to a realistic problem. It created a meaningful experience for the children, motivating them to explore and learn, which was key to children's learning (Assignment_Student#4).

> … makes it a bit more authentic; you can see it actually in front of you. It maybe caters for more learners so that they're tactile and kinaesthetic, as well as maybe the visual… But it's right in front of you and it's actually happening and you can manipulate it. So, it's learning on the go, it's about always progressing. (Focus Group 1)

> Later in this class, a model failed to move forward in a race that was being conducted. Here another child offered a very strong reasoning for this issue, 'well, if my chain falls off my bike, it doesn't go, and didn't you say the pulley is like a chain? It could be the pulley is after falling off of it'. This observation ratified for us that the children were beginning to not only learn from each other, but make connections with their reasonings from their models to real life. This was also complimentary of a valuable point previously made by Cszmadia based on collaboration with peers to "evaluate each other's code, to isolate bugs, and to suggest fixes." (Assignment_Student#6)

Prior to this classroom experience, the preservice teachers had been concerned as to how they could implement the "Digital Strategy for Schools" [34], with reference to the Digital Learning Framework [35] and accommodate the development of computational thinking into the curriculum. However, through their experiences of teaching with the computational materials, they realised the use of computational materials opened new ways of exploring scientific ideas while also providing opportunities for developing computational thinking. Having "an object to think with" as they tried to solve problems and challenges enabled the children they engaged with to develop understandings of science concepts and skills (e.g. inquiry, analysis, observing, predicting, experimenting and designing) but also computational thinking skills (e.g. algorithmic thinking, decomposition, debugging). As student teachers said:

> This research has shown how both the chosen DLF [Digital Learning Framework] outcome and CT [computational thinking] can be implemented through the science curriculum (Assignment_Student#16)

> … the teacher has a pivotal role in enabling children to develop this higher order skill-set, thus planning through subject integration is pivotal as suggested by the Irish DLF (Assignment_Student#11)

5 Discussion and Conclusion

To conclude, working with the children in the classroom helped the preservice teacher develop their own understandings of what computational thinking looked like "in action" and reflect more deeply on the fundamentals of computational thinking and on how to use computational thinking in their own classroom practice as qualified teachers. Their competence and confidence in using the computational materials developed and they were able to connect principles of constructionism and inquiry-based science with an informed understanding of the necessity to plan for the development of computational thinking in primary classrooms. Having the opportunity to engage in this module, rooted in classroom practice, enabled these preservice teachers understand, how computational thinking can be developed by embedding it within existing curricula, leading to the realisation that it is not a case of one or the other (i.e. computational thinking or subject content) but a means of combining both.

The insights gained from this study are particularly relevant for the design of teacher preparation programmes, indicating how computational thinking can be effectively embedded to combine theory and practice. This will ensure that concepts of computational thinking are not developed in a decontextualised manner but are embedded within the prescribed curriculum across a range of subject content in a relevant and meaningful manner.

References

1. Angeli, C., et al.: A K-6 computational thinking framework-implications for teacher knowledge. Educ. Technol. Soc. **19**(3), 47–57 (2016)
2. Fluck, A.E., et al.: Arguing for computer science in the school curriculum. Educ. Technol. Soc. **19**(3), 38–46 (2016)
3. Kafai, Y.: From computational thinking to computational participation in K-12 education. Commun. ACM **59**(8), 26–27 (2016)
4. ISTE, CSTA: Computational Thinking in K–12 Education Leadership Toolkit. Computer Science Teacher Association, https://id.iste.org/docs/ct-documents/ct-leadershipt-toolkit.pdf?sfvrsn=4. Accessed 05 Jun 2020
5. Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., Korb, J.T.: Introducing computational thinking in education courses. In: Proceedings of the 42nd ACM Technical Symposium on Computer Science Education, pp. 465–470. ACM, New York (2011)
6. Lye, S.Y., Koh, J.H.L.: Review on teaching and learning of computational thinking through programming: what is next for K-12? Comput. Hum. Behav. **41**, 51–61 (2014)
7. Barr, V., Stephenson, C.: Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? ACM Inroads **2**(1), 48–54 (2011)
8. Wing, J.M.: Computational thinking, 10 years later, p. 23. Microsoft Research Blog (2016)
9. DES: Irish Primary School Science Curriculum. Government Publications Office, Dublin, Ireland (1999)

10. Kafai, Y.B., Resnick, M. (eds.): Constructionism in Practice: Designing, Thinking, and Learning in a Digital World. Routledge, New York (2012)
11. Papert, S., Harel, I.: Situating Constructionism. In: Constructionism. Ablex Publishing, New York (1991)
12. Papert, S.: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, New York (1980)
13. Turkle, S., Papert, S.: Epistemological pluralism: styles and voices within the computer culture. Signs J. Women Cult. Soc. **16**(1), 128–157 (1990)
14. Butler, D.: A Constructionist View of what it means to be Digitally Literate Learning in 21st Century Classrooms. Nordic Journal of Digital Literacy **2**(02), 61–77 (2007)
15. Ostwald, J.: Knowledge construction in software development: the evolving artifact approach. https://dl.acm.org/doi/book/10.5555/923284. Accessed 03 Jun 2020
16. Ackermann, E.: Piaget's constructivism and papert's constructionism: what's the difference? In: Constructivisms: Usages et Perspectives en Education, vol. 1 and 2, pp. 85–94. SRED/Cahier, Geneva (2001)
17. Dede, C., Mishra, P., Voogt, J.: Working group 6: advancing computational thinking in 21st century learning. In: EDUsummIT 2013: International Summit on ICT in Education, San Bruno, CA (2013)
18. Wing, J.M.: Computational thinking. Commun. ACM **49**(3), 33–35 (2006)
19. Grover, S., Pea, R.: Computational thinking: a competency whose time has come. In: Computer Science Education: Perspectives on Teaching and Learning in School. Bloomsbury Academic, London (2018)
20. Selby, C., Woollard, J.: Refining an understanding of computational thinking. https://eprints.soton.ac.uk/372410/1/372410UnderstdCT.pdf. Accessed 05 Jun 2014
21. Voogt, J., Fisser, P., Good, J., Mishra, P., Yadav, A.: Computational thinking in compulsory education: towards an agenda for research and practice. Educ. Inf. Technol. **20**(4), 715–728 (2015)
22. Brennan, K., Resnick, M.: New frameworks for studying and assessing the development of computational thinking. In: AERA 2012, Vancouver, BC (2012)
23. Curzon, P., Dorling, M., Ng, T., Selby, C., Woollard, J.: Developing computational thinking in the classroom: a framework. https://eprints.soton.ac.uk/369594/. Accessed 05 Jun 2014
24. Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., Danies, G.: Changing a generation's way of thinking: teaching computational thinking through programming. Rev. Educ. Res. **87**(4), 834–860 (2017)
25. Hodhod, R., Khan, S., Kurt-Peker, Y., Ray, L.: Training teachers to integrate computational thinking into K-12 teaching. In: Proceedings of the 47th ACM Technical Symposium on Computing Science Education, pp. 156–157. ACM, New York (2016)
26. Lodi, M.: Growth mindset in computational thinking. teaching and teacher training. In: Proceedings of the 2017 ACM Conference on International Computing Education Research, pp. 281–282. ACM, New York (2017)
27. Yadav, A., Gretter, S., Good, J., McLean, T.: Computational thinking in teacher education. In: Rich, P.J., Hodges, C.B. (eds.) Emerging Research, Practice, and Policy on Computational Thinking. ECTII, pp. 205–220. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-52691-1_13
28. Zha, S., Jin, Y., Moore, P., Gaston, J.: Hopscotch into coding: introducing pre-service teachers computational thinking. TechTrends **64**(1), 17–28 (2019). https://doi.org/10.1007/s11528-019-00423-0
29. Cator, K., Angevine, C., Weisgrau, J., Waite, C., Roschelle J.: Computational thinking for a computational world. http://digitalpromise.org/wp-content/uploads/2017/12/dp-comp-thinking-v1r5.pdf. Accessed 05 Jun 2017

30. Teaching Council: Initial Teacher Education Criteria and Guidelines. https://www.teachingcouncil.ie/en/Publications/Teacher-Education/Initial-Teacher-Education-Criteria-and-Guidelines-for-Programme-Providers.pdf. Accessed 05 Jun 2017
31. Hatch, J.A.: Doing Qualitative Research in Education Settings. Suny Press, New York (2002)
32. Lu, J.J., Fletcher, G.H.: Thinking about computational thinking. ACM SIGCSE Bull. **41**(1), 260–264 (2009)
33. Csizmadia, A., et al.: Computational thinking - a guide for teachers. https://www.researchgate.net/publication/327302966_Computational_thinking_-_a_guide_for_teachers/link/5bcedae7299bf1a43d9a6276/download. Accessed 05 Jun 2015
34. DES: Digital Strategy for Schools: Enhancing Teaching, Learning & Assessment. https://www.education.ie/en/Publications/Policy-Reports/Digital-Strategy-for-Schools-2015-2020.pdf. Accessed 05 Jun 2015
35. DES: Digital Learning Framework and Planning Resources – Primary. https://www.pdsttechnologyineducation.ie/en/Planning/Digital-Learning-Framework-and-Planning-Resources-Primary/. Accessed 05 Jun 2002