Coding for non-Computer Scientists: Pedagogies for interdisciplinary participation

Georgie Tarling
Graduate School of Education
University of Exeter, UK
g.tarling3@exeter.ac.uk

Ana Isabel Melro
Graduate School of Education
University of Exeter, UK
Communication and Society Research Centre
University of Minho, Portugal
a.melro@exeter.ac.uk

Judith Kleine Staarman
Graduate School of Education
University of Exeter, UK
j.kleine-staarman@exeter.ac.uk

Taro Fujita
Graduate School of Education
University of Exeter, UK
t.fujita@exeter.ac.uk

Abstract: Coding is increasingly seen as a valuable subject to learn, with initiatives to promote the teaching of coding underpinned by three main rationales: employability, interdisciplinary problem solving and informed citizenship. These highlight the need for courses which teach more than just functional skills. To be meaningful, the learning of coding needs to be contextualized more holistically and focus on the development of computational thinking (CT) skills that are important for wider interdisciplinary participation.–In this paper, we propose a participation focused pedagogical approach for an Institute of Coding Summer School, whose purpose is to engage non-Computer Science students in learning the basics of coding, data analytics and artificial intelligence. Drawing on data from questionnaires, interviews, and observations we identify best practices and articulate these within a three-dimensional conceptual framework for CT. Our findings describe the skills in relation to coding these students think will have authentic, real-world value. We then explore in more detail certain pedagogical strategies used to create contextualised and real-world learning experiences alongside the value of collaborative coding. We conclude by considering the wider potential application of our findings to other computing education contexts.

Keywords: Coding, computational thinking, pedagogies, interdisciplinary participation

Introduction

Coding is increasingly seen as a valuable and relevant thing for non-computer science (CS) students to learn. This aligns with trends in mainstream international debate (Grover & Pea, 2017; Popat & Starkey, 2019), where initiatives to promote the teaching of coding are underpinned by a range of different rationales. Firstly, coding is seen as a vital skill for 21st century workers (Lockwood & Mooney, 2018) in the changing jobs landscape (Orlik, 2018; Ponce, 2018; Williamson, 2016; World Economic Forum, 2018). Secondly coding is seen as a part of a wider suite of 21st century transferable skills including team work and analytical thinking that are needed across all disciplines to bring diverse people together to solving the problems of the world (Burke et al., 2016; Grover & Pea, 2017; Popat & Starkey, 2019; Ryoo et al., 2013). Finally learning to code is seen as necessary for full participation as an informed citizen in the digital world (Bocconi et al., 2016; Grover & Pea, 2013), as it provides a foundation for creative and critical engagement with the affordances, limitations and assumptions embedded in code (Burke et al., 2016; diSessa, 2018; Dufva & Dufva, 2016). These rationales highlight that what learners need is more than just courses which teach the functional skills of coding. To be meaningful and useful, the learning of coding needs to be contextualized more holistically (Dufva & Dufva, 2016). This multitude of skills in the learning of coding is discussed in the literature around concepts of computational thinking (CT), where a range of frameworks and terminologies, including computational participation, illustrates the complexity and confusion of this subject (Brennan & Resnick, 2012; Burke et al., 2016; Gretter & Yadav, 2016; Grover & Pea, 2017). Adding to this, there is little evidence of coding courses being underpinned by CT theoretical frameworks and even less detailed evaluation of their pedagogies (Waite, 2017), particularly in meeting the real world needs of a diverse range of students.

With an intent to bridge existing concepts of CT and create a model that can guide pedagogical practices in coding courses, we developed a conceptual framework that encompasses three dimensions in the teaching of coding to non-CS students (Melro et al., 2020): (1) functional, (2) participatory and (3) critical and creative. Drawing on existing literature and student data from a coding course, this framework, and particularly the second dimension, highlights the importance of learning coding for interdisciplinary participation. At a macro level, the participatory dimension highlights how important it is for learners to engage in conversations around data science and Artificial Intelligence (AI), and to unveil further career and study opportunities that combine their background of expertise with CS. This is crucial to foster participation in coding communities and to participate in the public sphere (Dufva & Dufva, 2016; Grover & Pea, 2017; Iversen et al., 2018). At a micro level, the second dimension entails the relevance of collaborative work, for instance in developing communication skills for discussing alternative solutions for problem-solving (Barr & Stephenson, 2011; Burke et al., 2016). Given this framework, we address the need to identify best pedagogical practices that create meaningful and contextualised learning experiences that allow non-CS students to engage in interdisciplinary participation that could be relevant for their future lives and careers.

The study

In this paper, we focus on the pedagogical practices and strategies in the learning of coding that help promote interdisciplinary participation in non-CS diverse learners. For this purpose, we gathered data from an Institute of Coding (IoC) Summer School, whose purpose is to engage non-Computer Science university students in learning the basics of coding, data analytics and artificial intelligence. The stated aims of the IoC are for its courses to embed innovative pedagogies, to increase diversity by appealing to a broad range of students, and to develop business, technical and interpersonal skills in equal measure. The evolution of this course is iterative and research informed, with the design and delivery shaped collaboratively by lecturers and researchers.

The IoC Summer School ran for the first time in 2018 and is comprised of three learning blocks. Block 1 (B1) is a Coding Boot Camp, teaching students how to formulate and solve problems through the use of Python. Block 2 (B2) is about Social Data Analytics, exploring statistical concepts, data visualisation and social research methods using R. Block 3 (B3) offers a series of lectures and workshops on AI and machine learning. Based on findings from 2018, in 2019, the three-week course had an increased focus on the use of pedagogical strategies to promote contextualised, real-world learning. For the purpose of this paper, we will focus on the quantitative and qualitative data of the 2019 IoC Summer School (n = 36) that was collected through nine different questionnaires (pre and post-attitude, and programme evaluation surveys – PES, for each block), interviews with students (n = 14), lecturers, guest lecturers and Teaching Assistants (TA) (n = 9) and observations (n = 15) of key moments of the sessions. The majority of the students in our sample are females (57.1%) from Social Sciences and Arts and Humanities (77.8 %). The data was analysed using R and SPSS for statistical analysis and NVivo for content analysis. In this study, we used a mixed-method approach in which quantitative and qualitative findings from different tools were discussed and a comprehensive content analysis of the qualitative data[1] from both the student interviews and questionnaires was proceeded.

1 In the following findings, student quotations are followed by an ID number, the research tool where the data was collected – PRE for the attitude questionnaires before the course, INT for the interviews and PES for the evaluation survey for B1, B2 or B3 – student's gender identity, age and discipline.

Findings

Combining the qualitative data from student interviews and questionnaires, Error: Reference source not found displays the categories of content analysis which address the need for an interdisciplinary participation with coding in terms of value, pedagogies and learning outcomes.

Addressing the need for interdisciplinary participation with coding (*n* = 36)	*n* cases	%
Value of interdisciplinary participation for careers and studies	30	83.3%
General careers and opportunities	22	61.1%
Specific jobs and fields	22	61.1%
Pedagogies and strategies for interdisciplinary participation	*NA*	*NA*
Social bonding	15	41.7%
Pair/group work (peer programming)	23	63.9%
Contextualised and real-world learning	18	50.0%
Learning outcomes for interdisciplinary participation	24	66.7%

Table 1. Categories of content analysis of student interviews and questionnaires regarding interdisciplinary participation

In the qualitative analysis, we found that coding's perceived relevance for students' lives and careers was explained by its interdisciplinarity. In our study, almost all students (83.3%) wanted to learn coding because they were interested in combining in one way or another their current background of expertise with the field of computer science. In the category "value of interdisciplinarity for career and studies" displayed in Error: Reference source not found, students either referred to specific careers in mind, such as "data analyst" or as a complement to their studies revealing interest for instance in Biosciences, data journalism or AI and ethics. As this student reveals: "I hope to work in data science eventually and am starting a MSc in data science for policymaking in September. This course should help in the social data side of things" (ID6, PESB2, male, 22 years old, Politics/International Relations). Also important is the expectation students revealed before the course around wanting to learn coding in order to have conversations with developers or to work in interdisciplinary teams. Other students would be less specific and refer to a more generic interest in terms of the opportunities that learning coding could bring to their future career. In this regard, students mentioned the idea that they would become more employable and would open more possibilities in the jobs market due to the perceived importance of technology today.

In terms of supporting the learning of coding for interdisciplinary practice, the IoC Summer School used a range of pedagogical strategies. In the PES questionnaires, overall students found the course useful for their future lives and careers, rating a mean of $M = 3.6$ ($SD = 1.0$; $n = 56$), meaning they felt they could apply "quite a bit" of what they learned in their future careers. Part of this good rating is due to the value that most students (50%) placed on contextualised and real-world learning. This was accomplished in the course firstly by adopting a strategy of including industry speakers as guest lecturers who provided students with real-life problems and made them aware of a diverse range of applications, for instance, from political polling to dementia support. Secondly, it was promoted in the main lectures through the use of examples relevant to students' backgrounds and in the workshops through the use of tasks involving contextually relevant datasets ranging from weather forecasting to economic data. Finally, it was promoted through the inclusion each week of a group project, the briefs for which were anchored in real-world application. These included making a game or app, researching, creating and presenting a piece of data journalism and building a facial recognition classifier. These real-world inputs and contextualisation by lectures and guest lectures were highly appreciated by students and proved to create a meaningful learning that could be useful for their interdisciplinary careers.

Another pedagogical strategy that proved relevant for students' future participation in interdisciplinary workplaces, was the scaffolding and support for collaborative work. Much has been discussed in the literature about the importance of pair and peer programming in the learning of coding, for instance, in how it helps foster teamwork and communication skills so students would be better prepared not only to collaborate in teams and communities using coding but also to engage in conversations with developers (Burke et al., 2016, p. 374). At the IoC Summer School, collaborative work was supported by the following: through pair work during practical activities, through the previously mentioned group projects at the end of each week and through building on existing code and developing an understanding of online coding communities. As observed in Error: Reference source not found, more than half of the students (64%) valued peer programming through which they recognised a couple of helpful learning strategies. For instance, through peer programming, students highlighted the advantage of peer debugging in spotting errors together, the importance of peer problem-solving in listening and discussing alternative solutions for problems using coding, and more importantly, the value of peer learning in explaining things to each other in a way that they could understand, reinforcing learning and confidence levels.

The findings revealed that pair work was found more helpful over group work mainly because the latter was more difficult to manage in terms of task division and reaching agreements due to its size. Nonetheless, in group projects students also learned how to communicate and present their ideas to the audience of the classroom, which lecturers also considered in their final decisions of the winning groups. The cohesion of the groups and peer collaboration was also improved when students felt they were paired with colleagues with roughly the same level of experience or with someone they better connected or were more familiar with. For the latter, the findings revealed that social bonding had great impact in collaborative work with almost half of the students (42%) recognising its value. At the IoC Summer School, social bonding was encouraged through coffee and lunch breaks, social venues after class and through classroom discussions during sessions. These moments were seen important to facilitate participation and collaborative work, as this student mention: “social events were really handy and useful for fostering greater participation in class (…)” (ID30, PESB2, female, 21 years old, Politics/International Relations). A broader way of developing students’ sense of collaborative coding was through teaching them how to build on and remix others’ code, use packages and libraries and draw on coding communities for support. This sense of coding as collaborative came as a surprise to some students. One commented that he hadn’t realised “the classifiers like the k-means that they spoke about in that course, you can use that and someone’s done a lot of the leg work, a lot of the hard work that goes on behind the scenes” (ID37, INT, male, 34 years old, Education). However, although the lecturers saw these as valuable skills the students were developing, the students themselves did not always understand that this was a learning outcome in its own right.

In terms of participatory skills in relation to coding that students perceived developing during the course, these can be categorised in two main areas; teamwork and awareness of application. Students felt that as a result of the course they were better equipped, both to work with others to solve problems using coding and also to have conversations with developers, as expressed in this comment: “[The block was] good for understanding the basics for interactions with developers” (ID12, PESB1, male, 24 years old, Economics/Business). They also felt better informed about the potential applications of coding and were starting to see how they might use coding in their further learning or careers. As one student noted: “The academic component is important, but it all comes to down to the practical application of it, how does it change lives, how does it improve the economy, how does it improve security, so that is the aspect that I was most interested in” ((ID20, INT, male, 34 years old, Social Sciences). By understanding the principles behind things like facial recognition, they felt more able to see how they might bring their own domain knowledge to bear on technology development.

Discussion and conclusions

Interdisciplinarity was found to be the most important reason why non-CS students learn coding either because they want to combine their studies or careers with CS or to engage in conversations with interdisciplinary teams. Interdisciplinarity in coding is also a major topic of discussion around data science careers since coding is seen as a vital skill for diverse people (Lockwood & Mooney, 2018). Thus, discussions around CT highlight the importance of approaching the teaching of coding in a way that not only prepares learners for the current interdisciplinary workplace but also promotes meaningful contextualised learning for wider participation (Burke et al., 2016; Melro et al., 2020). Drawing on data from an introductory coding course, in this paper, we were able to identify a set of important strategies and pedagogies that proved to be helpful by diverse learners in addressing a wider interdisciplinary participation.

To sum up, Error: Reference source not found gathers the strategies and pedagogies that were found useful in the data from the 2019 IoC Summer School and their relationship with the range of skills or components of both macro and micro levels of the participatory dimension of CT (Melro et al., 2020). It is important to note that the micro level skills or components are key elements that contribute to a macro level. In other words, learning communication skills, how to work collaboratively using coding, how to negotiate with colleagues and improve interpersonal relationships are essential to a wider context of working in interdisciplinary teams or jobs, engaging in conversations with developers and participating in coding communities. These skills and components focus on the value of working with “others” or in developing a ‘conversational’ viewpoint in learning and working with coding.

These are developed in the Summer School through peer programming, remixing codes, fostering social bonding, engaging in real-world activities and understanding real applications of coding within their previous backgrounds.

Participatory dimension of CT	**Skills/components**	**Strategies/pedagogies**
Macro level	Interdisciplinarity between fields Interdisciplinary teamwork Participation in coding communities	Contextualisation of lectures and activities Industry speakers Remixing and sharing code using repositories and engaging in online coding communities Group/pair work Peer programming (peer problem-solving, peer debugging) Group projects (presenting things to others, real-world problems) Icebreakers, social venues
Micro level	Collaboration Communication Negotiation Social bonding	

Table 2. Strategies and pedagogies for interdisciplinary participation

To conclude, the participatory dimension fits into our wider conceptual framework for CT which also gives value to functional and creative/critical dimensions (Melro et al., 2020). This framework perceives learning in each of these dimensions as layered: the skills developed in one build upon and consolidate the learning of coding in the others. Whilst acknowledging the limitations of our study which focuses only on non-CS students, we suggest that this more holistic contextualised, real world approach offers a meaningful way of learning coding that could also prompt some reconsideration of the teaching of Computer Science students to include broader skills. This might address the so-called 'digital skills gap' in employment rates for Computer Science undergraduates in the UK (Davenport et al., 2019). Further research is needed about the practical applications of this framework and its meaningfulness in wider educational coding contexts.

References

Barr, B. V., & Stephenson, C. (2011). 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.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). JRC Science for Policy Report. Developing Computational Thinking in Compulsory Education: Implications for policy and practice. In *European Commission* (Issue June). https://doi.org/10.2791/792158

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *American Educational Research Association Meeting (AERA)*. https://doi.org/10.1007/978-3-319-64051-8_9

Burke, Q., O'Byrne, W. I., & Kafai, Y. B. (2016). Computational Participation. *Journal of Adolescent & Adult Literacy*, *59*(4), 371–375. https://doi.org/10.1002/jaal.496

Davenport, J. H., Crick, T., Hayes, A., & Hourizi, R. (2019). The Institute of Coding: Addressing the UK digital skills crisis. *ACM International Conference Proceeding Series*, 0–3. https://doi.org/10.1145/3294016.3298736

diSessa, A. A. (2018). Computational Literacy and "The Big Picture" Concerning Computers in Mathematics Education. *Mathematical Thinking and Learning*, *20*(1), 3–31. https://doi.org/10.1080/10986065.2018.1403544

Dufva, T., & Dufva, M. (2016). Metaphors of code—Structuring and broadening the discussion on teaching children to code. *Thinking Skills and Creativity*, *22*, 97–110. https://doi.org/10.1016/j.tsc.2016.09.004

Gretter, S., & Yadav, A. (2016). Computational Thinking and Media & Information Literacy: An Integrated Approach to Teaching Twenty-First Century Skills. *TechTrends*, *60*(5), 510–516. https://doi.org/10.1007/s11528-016-0098-4

Grover, S., & Pea, R. (2013). Computational Thinking in K–12. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Grover, S., & Pea, R. (2017). Computational thinking: A competency whose time has come. *Computer Science Education: Perspectives on Teaching and Learning in School*, *December*, 19–39. http://hub.mspnet.org/index.cfm/33300

Iversen, O. S., Smith, R. C., & Dindler, C. (2018). From computational thinking to computational empowerment. *PDC '18, August 20–24, 2018, Hasselt and Genk, Belgium*, 1–11. https://doi.org/10.1145/3210586.3210592

Lockwood, J., & Mooney, A. (2018). Computational Thinking in Secondary Education: Where does it fit? A systematic literary review. *International Journal of Computer Science Education in Schools*, *2*(1), 41. https://doi.org/10.21585/ijcses.v2i1.26

Melro, A., Tarling, G., Fujita, T., & Kleine Staarman, J. (2020). Thinking about Computational Thinking: non-Computer Science students' perceptions of coding and reasons for learning how to code. *(Accepted) Computer Science Education*.

Orlik, J. (2018). *Delivering digital skills: A guide to preparing the workforce for an inclusive digital economy*. 48. https://media.nesta.org.uk/documents/Readie_Digital_Skills_booklet_online.pdf

Ponce, A. (2018). Artificial Intelligence: A Game Changer for the World of Work. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3198581

Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers and Education*, *128*(October 2018), 365–376. https://doi.org/10.1016/j.compedu.2018.10.005

Ryoo, J. J., Margolis, J., Lee, C. H., Sandoval, C. D. M., & Goode, J. (2013). Democratizing computer science knowledge: Transforming the face of computer science through public high school education. *Learning, Media and Technology*, *38*(2), 161–181. https://doi.org/10.1080/17439884.2013.756514

Waite, J. (2017). Pedagogy in teaching Computer Science in schools: A Literature Review. *The Royal Society*, 1–90. https://royalsociety.org/~/media/policy/projects/computing-education/literature-review-pedagogy-in-teaching.pdf

Williamson, B. (2016). *The learning to code and digital making movement*. Code Acts in Education Research Summary 3. https://codeactsineducation.wordpress.com/

World Economic Forum. (2018). *The Future of Jobs Report 2018*. World Economic Forum - Centre for the New Economy and Society.

Acknowledgements

This work was supported by the Institute of Coding which received funding from the Office for Students (OfS).