This is the author's accepted manuscript of:
Shelton, C. (2024) 'Supporting Lower Attaining Pupils in Early Computing Education.' In Ed. Holbert, N. and Blikstein, P. (2024) Proceedings of Constructionism / Fablearn 2023 Carnegie Mellon University Press p103 – 111.
ISBN: 978-1-300-96915-0. Doi: 10.57862/jfps-mb68.
Available at https://press.etc.cmu.edu/proceedings/proceedings-constructionism-fablearn-2023

Supporting Lower Attaining Pupils in Early Computing Education

Chris Shelton

University of Chichester, UK, c.shelton@chi.ac.uk

Learning computing and computational thinking is now a mandatory part of the primary curriculum of many countries and there are a range of different approaches to introducing the subject to young children. These early experiences of computing are important for motivating and engaging pupils and, at their best, can spark a life-long interest in the subject. However, some pupils struggle to learn computing and computer science and risk becoming disengaged – avoiding this happening is a key challenge for school teachers. This paper explores some pedagogical features that show promise for enabling all pupils to succeed in computing, particularly those who find it most challenging. The paper discusses a case study of computing education for pupils aged 7-9 where the pupils were taught using the constructionist 'Computing with Emil' scheme. It cites observation, interview, work sample and formal assessment evidence to discuss features of the approach that were most successful in helping those pupils who struggle the most to learn computing. These included setting accessible and motivating activities, providing opportunities for collaboration, ensuring frequent experiences of success, and using multiple methods of recording.

CCS CONCEPTS • K-12 Education • Computational Thinking

Additional Keywords and Phrases: attainment; inclusion

1 INTRODUCTION

> "One of the challenges (but also the pleasures) of developing a totally inclusive learning environment within the classroom is that, by their very nature, all children are different and have different strengths and needs" (Langston, p339, [1])

Ensuring the progress of all learners in their class has been described as "by far the greatest challenge to teachers" [2] and this is as true for computing as any other subject. The early years of schooling are crucially important for children and early experiences can have a long-lasting influence on a child's education and learning. With computing now a feature of mandatory primary education in many jurisdictions, it is important that early experiences of the subject are positive and allow all children to learn and be successful. This is particularly vital as continuation rates for young people taking computing-related subjects beyond compulsory schooling remain low in many countries, including the UK, and this is particularly the case for those from disadvantaged and under-represented groups.

In order for pupils to have a positive experience of learning computing, teachers need to recognize and cater for the different experiences, strengths and needs of the children in their class. However, teachers can find this challenging [3] and teacher perceptions and beliefs about pupils' abilities are often gendered and biased [4]. This paper discusses one small aspect of inclusive computing teaching: how teachers can ensure that they support the learning of the lowest attaining pupils in their class. There is currently a lack of research and guidance for teachers on how they might to do this and the paper aims to identify some potential avenues for future exploration.

2 ATTAINMENT IN COMPUTING

In any computing class, there will be differences in pupils' levels of attainment [5] although it can be argued that computing can be a "good subject for reaching out to learners who may not achieve in other areas" [6]. However, we must be careful not to assume that pupils' attainment in computing is necessarily due to a lack of aptitude or 'ability'. In fact, differing levels of attainment may just reflect how much opportunity pupils have had to experience computing outside of school and this in turn will be influenced by a range of social and cultural factors. Differences in attainment can be related to cultural capital or home language [7] or reflect wider social disparity and under-representation. In addition, teachers can hold an unconscious bias that certain students cannot be as successful in computing, for example, due to a disability [8].

School computing teachers are very conscious of the challenges and difficulties that they encounter. In a survey of over 300 school teachers, Sentance and Csizmadia [3] identified that teachers found the range of attainment within a typical class challenging. For example, some teachers felt that their students' prior experience of programming differed and that this was hard to cater for with one teacher quoted as saying "I have found the ability gap to be much bigger than any other subject" (p480). Other teachers referred to these gaps between children's attainment widening as some student progress faster than others. The teachers in this study were able to identify specific aspects of computing that their pupils struggled with and these included both concepts (e.g. variables) and approaches (e.g. problem-solving and thinking computationally).

Matching the material to be learnt to the needs and characteristics of learners can be considered an aspect of teachers' pedagogical reasoning. Webb identifies several stages to pedagogical reasoning in teaching computing including selecting teaching strategies, adapting these to the characteristics of students, and then tailoring those to the needs of individuals [9]. While some strategies for matching computing work to the needs of individuals have been proposed, for example, allowing pupils to self-select from a range of different challenges [10], teachers are not always well equipped to ensure that they set work that is at the right level of challenge for all their pupils and even experts can find it difficult to evaluate the complexity and difficulty of computing tasks [11].

Instead of setting different levels of activity, teachers can also introduce learning opportunities that are designed to be accessible to all pupils (having a 'low floor') while allowing pupils to demonstrate high levels of achievement (having a 'high ceiling') [12]. Such designs allow all pupils to develop agency, to engage with computing concepts and to take part in programming activities. However, it is important that activities for young children promote both thinking and doing, rather than just doing [13]. One approach that aims to encourage pupils to think deeply about computational concepts is Computing with Emil.

Computing with Emil is a programming environment and systematic pedagogic approach for lower primary computing [14, 15]. The Computing with Emil approach includes a programming microworld consisting of carefully designed gradations of tasks; a pupil workbook; and a set of teacher materials. Pupils work in pairs using a laptop or tablet to complete tasks in the microworld and work individually to complete tasks in the workbook. A key principle underpinning the design of the programme is that "there is no space for lecturing" [14], learning takes place through pupils collaborating, completing tasks and through whole class discussion scaffolded by the teacher. The software does not provide feedback so that pupils are encouraged to discuss and agree whether or not a task has been solved [15]. In the Computing with Emil 3 unit used in this research, the key concepts covered are sequence, order and coping with constraints.

3 METHODS

The examples used in this paper are taken from a research project exploring the use of the Computing with Emil within the context of the National Curriculum in England. This research was conducted in England during the 2022-23 academic year and this paper draws on data from the project that relates to the lowest attaining pupils in the case study class.

The research was conducted in an English Primary School with a class of 26 pupils aged 7-9. As is usual in UK schools, the school year is divided into six 'half-terms' and pupils learn different topics each half-term. For the purposes of this project, the school agreed to teach the Computing with Emil 3 curriculum and this paper reports data collected from a half-term of lessons – this consisted of 6 lessons each lasting between 40 - 50 mins. The lessons covered activities from the 'World 1 - Emil the Collector' unit which focus on concepts of sequencing and order. As the unit progresses, these activities introduce various forms of constraint that pupils learn to appreciate and, sometimes, overcome.

Prior to the start of the Computing with Emil lessons, an initial assessment of computational thinking was conducted using the TechCheck 2 Computational Thinking assessment [16]. This is a validated, unplugged assessment tool that uses multiple choice questions that require no prior knowledge of computer programming. The question style does not resemble the style of questions experienced in the Computing with Emil microworld or workbook.

The TechCheck assessment is based around Bers "Seven Powerful Ideas" [17] of developmentally appropriate computational thinking: algorithms, modularity, control structures, representation, hardware/software, design process, debugging. As a multiple-choice assessment, TechCheck assesses six of these computational thinking domains, omitting 'design process' as this "is an iterative and open-ended process with many solutions that cannot be readily assessed with the short multiple-choice format" [16]. Validation studies of TechCheck with pupils of similar age to those in this research sample had an average score of 11.58 [16].

During the lessons, data was collected through observational notes (including transcription of pupil speech). These notes included observations of: pupils using the Computing with Emil software; pupils working in pairs; pupils completing workbooks; contributions to whole class discussions; interactions between pupils and teachers. Copies were also made of the exercises that pupils completed in their workbooks.

After the half-term sequence of lessons was completed, an interview was conducted with the class teacher – this was recorded and transcribed for analysis. In addition, a second Tech Check Computational Thinking assessment was conducted and compared to the results from the initial assessment.

The research study followed the British Educational Research Association's "Ethical Guidelines for Educational Research"[18] and full ethical approval was given through the researcher's institution's ethical approval procedure. The headteacher of the host school and the pupils' class teacher were fully informed of the purpose of the research prior to the start of the study and both gave written consent. The participating teacher was able to withdraw from the research at any time during the study. Participation in the lessons was a compulsory part of the school curriculum for pupil but children were free to withdraw from the formal assessments and gave oral assent to these on both assessment occasions.

4 RESULTS

4.1 Pupil learning

Over the course of the project, the pupils had just six lessons using the Computing with Emil curriculum but despite the minimal amount of time devoted to computing, pupils made clear progress in their understanding of the concepts intended to be learnt through the Computing with Emil syllabus and in the development of their computational thinking. This was evident in the pupils' workbooks where regular tasks showed how pupils were able to respond to more complicated tasks and were able to cope with multiple constraints. It was also clear to the pupils' teacher who remarked on pupils' learning in a range of areas including "logical thinking" and "critical thinking".

The progress in computational thinking was demonstrated most clearly in the Tech Check assessments. Due to pupil absence on either the first or second assessment date, 21 pupils attempted both the initial and final assessments. The average score in the initial assessment was 9.9 out of 15 and the average score in the final assessment was 11.5 out of 15. Within these, the greatest improvements were made by those pupils with the lowest initial score, for example, Pupil H made a 6 point improvement. Increases could be seen against all of the six computational thinking domains measured by the TechCheck assessment (the greatest improvement was seen in the 'debugging' domain – however, due to the small number of questions in each domain and the size of the sample, differences between domains should not be overstated).

It would not be justifiable to claim that the measurable improvements seen over even this short period of time could be attributed to any one feature to the approach. However, the data did identify several features that appeared to be significant for some pupils and are worth further investigation, these included: accessibility of learning activities; collaborative work; frequency of success; and use of written tasks.

4.2 Access to learning

The class teacher described the Computing with Emil approach as being "so much more accessible" and "child friendly" than the other computing schemes that she was familiar with. She explained that this was because pupils were able to start working on problems immediately and then ask themselves questions about why something they had tried didn't work.

The ease of use and motivational potential of the software was also noted in observations – pupils were very enthusiastic about the lessons and were excited to take part in them. The graduated series of tasks through out the microworld ensured that pupils had a high level of success at the start of each sub-topic. This promoted pupils' independence and enjoyment of the activities.

The design of the activities and the teacher guide ensured that the materials were also accessible and easy to use for teachers. One of the difficulties for generalist teachers when teaching computing can be a lack of detailed subject knowledge in computing. In the interview, the class teacher remarked that although she would rate her computing knowledge as "7/10", her usual scheme of work was "almost too hard for my subject knowledge as a teacher…I find it really difficult" and that this constrained her ability to engage pupils in her lessons. In particular, she remarked on the "jargon" of computer science and not immediately knowing the answers to questions that pupils asked. In contrast, she believed that the Computing with Emil lessons supported her to teach effectively.

4.3 Collaborative work

The class teacher interview and lesson observations both noted that during the paired work, pupils frequently demonstrated sustained thinking and in-depth conversations about the tasks they were working on. The class teacher identified that one of the benefits of the Computing with Emil lessons had been the pupils' learning about working collaboratively. In pairs, they questioned each other, found different answers to problems and found ways of agreeing or disagreeing with their friends. This was the case for all pupils including those who were not meeting the age-related expectations in other areas of the curriculum.

Observations showed that the paired activities also allowed pupils to support each other. In one example, recalled by the class teacher, a pupil became frustrated when he was able to complete a task in multiple ways while his partner could not. Within the Computing with Emil session, he was able to guide his partner with open questions but without just telling him the answer. The teacher suggested that this was notable because the pupil had not been able to demonstrate this ability in other lessons such as mathematics.

Observations of Pupil E, for whom English was not their first language, suggest that she was most successful when working with support whether that was from paired partner, or in one lesson, an adult scribe.

4.4 Frequent success

Computing with Emil is designed around gradations of tasks and pupils complete multiple tasks in each lesson through the microworld and/or their workbook. The small challenges were described by the class teacher as providing frequent experiences of success so that pupils were "kept on board". In the teacher's words, pupils felt "We've achieved this. Now we move on. This is a bit of a harder task, [we] achieve it again and we're getting a lot more confidence and feeling like 'oh actually I can do this'." She also suggested that these short activities enabled pupils to "challenge themselves at their own level". This was also supported by the Teachers' Guide that suggested extension challenges that could be set for children to further deepen their understanding of a particular concept or to apply their knowledge in a more challenging context.

This experience of frequent success built confidence even for pupils with additional needs – Child S had struggled to engage with the initial TechCheck assessment, rushing through quickly without taking time to think about his response. He achieved much more highly in the final assessment and was observed considering his answers carefully. The class teacher explained that this pupil would often refuse to sit tests in other subjects and that this improved engagement reflected his motivation throughout the lessons.

Observations also noted that there was occasionally a difference in how different groups of pupils tackled the small exercises. Some exercises asked pupils to find multiple solutions to a particular task and while higher attaining pupils tended to follow the task instruction, lower-attaining pupils were observed finding a single answer and then rushing onto the next problem. This may require careful monitoring by the class teacher if finding multiple solutions is a key objective of the lesson.

4.5 Use of written tasks

The class teacher noted that as the Computing with Emil lessons did not require a large amount of writing, this helped cater for pupils in the class with special educational needs or who spoke English as an additional language. Sometimes, however, even minimal written activities can pose an additional challenge for learners. For example, one pupil who was observed demonstrating their understanding of sequencing in the microworld was unable to record their answer to the problem accurately in the workbook. However, the class teacher believed that the level of literacy required by the Computing with Emil scheme was lower than that of her usual computing scheme. She suggested it used less technical terms and was less reliant on pupils needing to "infer lots of different skills".

The Computing with Emil workbooks provide a rich source of evidence of pupils' understanding and learning over the sequence of lessons. They demonstrate how pupils used different methods to solve problems and, as the sequence of written tasks become gradually more challenging, the workbooks allow pupils to demonstrate their knowledge and teachers to identify pupil difficulties or misconceptions. The use of the workbook was an integral part of the Computing with Emil method and pupils came to appreciate how this helped them learn. For example, in one observation a pupil who was finding a problem in the microworld difficult, stopped using the computer and started to draw routes in their workbook, explaining to the observer that this 'helps me think'.

In the case of lower attaining pupils, written work in Computing could provide misleading information about pupil understanding if deficiencies in literacy lead to the pupils reading instructions incorrectly or being unable to express themselves accurately in writing. The workbook tasks include a variety of methods of gathering pupils' responses that do not require writing including drawing paths on screen-captured images and multiple-choice tick box responses. In addition, even when writing is required, simple instructions and images can make it easier for pupils to demonstrate their understanding even when their literacy levels are a barrier to their learning.

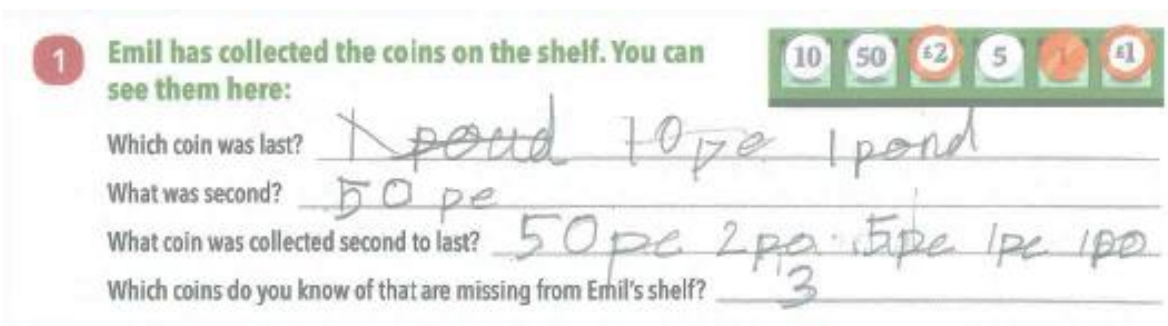

Figure 1: Workbook extract – Pupil N

Figure 1 is a single question from the Computing with Emil workbook. This was the first question in the workbook and so the pupils' first experience of the workbook question style. The task was set after pupils had completed the first set of Computing with Emil activities on their computers. These activities introduced the Robot Emil system and gave the pupils experienced of gathering ordered and unordered sets. In this question, the pupil had to interpret the image of Emil's "shelf" and use this to infer the order in which Emil had collected the coins.

Pupil N had scored 6 in the initial TechCheck assessment putting them in the lowest 25% of the class. However, as the task above shows, Pupil N had understood that the order on the shelf matches the order in which the robot collected the coins and could use this to identify the last and the second coins to be collected. As this was one of the objectives of the first sequence of activities, Pupil N can be seen to have achieved this. But Figure 1 also shows that Pupil N took three attempts to answer the first question – appearing to change only the spelling rather than their answer. In fact, the pupil's difficulties with literacy are clearly demonstrated in the question responses not only with the three (incorrect) spellings but also in the unconventional "pe" for "pence" (usually represented "p") and further misconceptions can be seen in the final question where Pupil N seems to suggest that a '3p' coin is missing – it is not there but no such coin exists.

The third question is also incorrect. When asked for the coin collected second to last, Pupil N has listed all the coins collected **from** the second one to the last one. This might suggest that the term 'second to last' is unfamiliar to the child rather than they unable to logically deduce which coin is collected immediately before the final one.

5 CONCLUSIONS

The relatively small amount of time devoted to learning computing over this study is typical of the amount of time that a primary class in England might spend on a computing topic. Despite this, there is evidence that the pupils made progress in their understanding of sequencing in computing and in their computational thinking. As the TechCheck assessment questions were not in the style of the Computing with Emil activities, this might suggest some element of 'near transfer' of learning from one computational thinking context to another. Also, while the focus of these Computing with Emil lessons were most closely related to the algorithms and debugging domains of the TechCheck assessment, there were increases across all domains of Computational Thinking and this requires further investigation.

The study demonstrates that the Computing with Emil approach does allow teachers to support the full range of levels of attainment in their class including those pupils with the lowest prior attainment. This may be due to several features of the constructionist approach including the accessibility and appropriateness of the activities for pupils and for teachers; the support provided through collaboration and paired activities; the opportunities for frequent success; and that written tasks use multiple forms that minimise the need for higher-level literacy skills.

In particular, the activities demonstrated a 'low floor'[12] that allowed all pupils to engage with computational concepts and provided multiple ways for pupils to achieve and demonstrate their learning. It was particularly successful in providing opportunities for thinking deeply as well as doing [13] and engaged pupils in 'hard fun' [19]. The large number of graduated tasks and the fact that the software does not indicate whether a particular task has been completed successfully served to slow down the pace of lessons allowing pupils to engage with ideas and concepts more deeply. Clearly designed gradations of tasks also help to support teacher subject knowledge by scaffolding their understanding of progression and difficulty in computing.

If teachers are going to ensure that all pupils, including those with lower prior attainment, are able to learn computing effectively, then future curriculum and pedagogic approaches should be designed to achieve this. Further research is needed to explore how these features of the Computing with Emil approach might be adapted in other approaches to teaching computing but it seems likely that providing time and space for pupils (and teachers) to engage with computational concepts and knowledge will be more effective that approaches that are more focused on introducing technical vocabulary and introducing new concepts quickly without time for consolidation and practice.

In addition, teachers need to be aware that pupils' literacy difficulties can lead to incorrect answers to computing tasks and that they should not assume that these necessarily imply errors of logical thinking or reasoning. Using a range of approaches to collecting written responses, such as drawing paths or multiple-choice responses, are less likely to be influenced by pupils' lower levels of literacy.

REFERENCES

[1] Langston, D. 2022. How do we ensure that our classrooms are truly inclusive? In Carden, C. Primary Teaching: Learning and Teaching in Primary Schools Today. 2nd Ed. Learning Matters.

[2] Titchmarsh A. 2023. Pupil Grouping, progression and adaptive teaching. In Capel et al. Learning to teach in the secondary school 9th Ed. Routledge. p210

[3] Sentance, S., Csizmadia, A. 2017. Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Educ Inf Technol* **22**, 469–495. https://doi.org/10.1007/s10639-016-9482-0

[4] Downes, T. and Looker, D. (2011) Factors that influence students' plans to take computing and information technology subjects in senior secondary school. Computer Science Education. 21(2): p. 175-199.

[5] Shelton C. 2017. How Can We Make Computing Lessons More Inclusive? In: Tatnall A., Webb M. (eds) Tomorrow's Learning: Involving Everyone. Learning with and about Technologies and Computing. WCCE 2017. IFIP Advances in Information and Communication Technology, vol 515. Springer, Cham. pp. 506-514 https://doi.org/10.1007/978-3-319-74310-3

[6] Caldwell, H. and Grantham, S. 2021. Computing. In Sewell, K. Planning the Primary National Curriculum 3rd Ed. p196

[7] Hatlevik, O.E. and Christophersen, K.-A. 2013. Digital competence at the beginning of upper secondary school: Identifying factors explaining digital inclusion. Computers & Education. 63: p. 240-247.

[8] Ladner, R.E. and Israel, M. 2016. "For all" in "computer science for all". Communications of the ACM. 59(9): p. 26-28.

[9] Webb, M. 2002. Pedagogical Reasoning: Issues and Solutions for the Teaching and Learning of ICT in Secondary Schools. Education and Information Technologies 7:3, 237–255.

[10] Shelton, C. 2016. Beyond lesson recipes: first steps towards a repertoire for teachng primary computing. In Proceedings of Constructionism 2016. Bangkok.

[11] Dagiene, V. and Stupuriene, G. 2016. Bebras - a Sustainable Community Building Model for the Concept Based Learning of Informatics and Computational Thinking. Informatics in Education. 15(1): p. 25-44.

[12] Flannery, L. P., Silverman, B., Kazakoff, E. R., Bers, M., Bontá, P., & Resnick, M. (2013). Designing ScratchJr. Proceedings of the 12th International Conference on Interaction Design and Children. https://doi.org/10.1145/2485760.2485785

[13] Palmér H. 2023. Children (Aged 3– 5 Years) Learning Mathematics Through Programming, Thinking and Doing, or Just Doing? In Ed. Keane, T. and Fluck, A. Teaching Coding in K-12 Schools: Research and Application. Springer International Publishing AG.

[14] Kalas, I. 2018. Programming in lower primary years: design principles and powerful ideas. In Proceedings of Constructionism 2018

[15] Kalas, I., Blaho, A., Moravcik, M. 2018 Exploring Control in Early Computing Education. In: Sergei N. Pozdniakov and Valentina Dagienė (eds.) Informatics in Schools. ISSEP 2018. LNCS, vol 11169, pp. 3–16. Springer, Cham.

[16] Relkin, E., de Ruiter., L., Bers, M. U. 2020. TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education. *Journal of Science Education and Technology.* https://doi.org/10.1007/s10956-020-09831-x

[17] Bers, M. U. 2018. Coding as a playground: programming and computational thinking in the early childhood classroom. Routledge. https://doi.org/10.4324/9781315398945 .

[18] British Educational Research Association [BERA]. 2018. Ethical Guidelines for Educational Research, fourth edition, London. https://www.bera.ac.uk/researchers-resources/publications/ethical-guidelines-for-educational-research-2018

[19] Papert, S. 2002. Hard Fun. Available at: http://www.papert.org/articles/HardFun.html