



Petrified of breaking things... and frightened of looking stupid: exploring ways to improve the teaching of computing in a UK Primary School.

Gemma Rolfe, BSc (Hons.) MSc PGCE PGCAP
Student ID No: 14223990

Thesis submitted to the University of Nottingham for the
degree of Professional Doctorate in Education.

September 2025

Abstract

The introduction of computing to the English National Curriculum in 2014 brought significant challenges for teachers. The teaching of computing, specifically Computational Thinking (CT), to primary pupils (ages 5-11) was argued to lead to long-term benefits for pupils, including increased problem-solving skills and greater participation among underrepresented or minority groups. However, many teachers did not feel prepared to teach CT or thought that the curriculum was inappropriate.

This research addressed this challenge in the specific case of a one-form-entry primary school in the North West of England. Starting three years after the introduction of the curriculum, the first phase of the study aimed to determine how the curriculum changes had been implemented in the school. Through document analysis, interviews and lesson observations of teachers and teaching assistants, it was found that the school had initially struggled with the change. This was evident in its translation of the National Curriculum objectives into the school curriculum, with further challenges arising during implementation. Teachers struggled specifically with the computer science or computational thinking (CT) parts of the curriculum. Computing was not taught consistently throughout the school year or across the different age groups within the school. While pupils were enthusiastic about computing, teachers reported challenges related to outdated equipment, planning that was

not fit for purpose, and a fear of not knowing how to teach computing, which was exacerbated by the lack of time to research the subject or integrate it into an already busy school timetable.

The second phase of the research aimed to develop and evaluate an intervention, through Continuing Professional Development (CPD), that would change how computing was perceived and taught in the school using a pedagogy-focused CPD that was context-specific and driven by middle leaders. Working with a change agent, a design-based research approach was adopted, enabling the refinement of curriculum design and continuing professional development based on the school's priorities and the change agent's experiences. Initially, working just with the change agent, the curriculum was rewritten to ensure complete coverage of the computing objectives, supporting progression in CT across the year groups. It was cross-curricular where possible, and key language was identified for teaching. CPD, including joint lesson planning, model lessons and team-teaching, had a positive impact on the change agent's subject knowledge and perceived confidence. The change agent's experiences of and outcomes from rewriting the curriculum and CPD led to the design of CPD for the rest of the teachers in the school. There were initial signs that this was positive; the teacher's medium-term planning showed an increase in computing objectives, and staff were keen to further develop their computing teaching. However, the

limitations imposed by COVID-19 adjustments and their implications unfortunately disrupted the longer-term follow-up of its impact.

This research concluded that changes to curricula without support for schools can pose challenges for teachers; however, working collaboratively with a passionate middle leader or change agent within a school can help alleviate these challenges. It is hoped that these findings will enable this research to contribute to knowledge and professional practice.

Acknowledgments

Thank you to my supervisors, Professor Shaaron Ainsworth and Dr Pryce Davis, for your consistent support and guidance throughout my EdD. I would not have reached this point without you both, and I am sorry it took so long!

I would like to express my gratitude to the staff and children of The Watermill School (pseudonym), who kindly agreed to my request to conduct research at their welcoming school. Special thanks to their amazing computing coordinator, Alex (also a pseudonym), for his enthusiasm for the computing curriculum and willingness to engage in change.

Thank you to all the brilliant trainee teachers and lecturers I have worked with over the last decade, especially those students who took the time to share their experiences of how computing was taught during their placements, which inspired this research.

To my family and friends who have been with me on this journey, especially my parents and brother, who have listened to my going on about computing and my boys, who will be glad to have Mummy back at the weekend!

Finally, completing a doctorate alongside full-time employment and adding two small beings to the mix would not have been possible without the support of my partner, Sam. Thank you for everything.

Table of Contents

CHAPTER 1: Introduction	1
1.1 Introduction.....	1
1.2 Overview/ Personal Context	1
1.3 Computing in the English Education System	2
1.3.1 Computational Thinking in Primary Education	10
1.3.2 The Role of Subject Leadership in English Primary Schools.....	16
1.4 Context	18
1.4.1 The School.....	18
1.4.2 The Staff	21
1.5 Thesis Goals	28
1.6 Layout of thesis	29
CHAPTER 2: Literature Review	31
2.1 Introduction	31
2.2 Computational Thinking (CT) in Education	31
2.2.1 Why should Computational Thinking be taught?	32
2.2.2 When should Computational Thinking be taught?	36

2.2.3 How are children taught Computational Thinking in schools?	38
2.2.4 What are teachers' perceptions of teaching computing and Computational Thinking?	48
2.3 Factors that prevent the teaching of computing and the use of educational technology	51
2.3.1 First-order Factors Affecting Technology Integration.....	52
2.3.2 Second-order Factors Affecting Technology Integration	56
2.3.3 The impact of teachers' self-efficacy on computing teaching	58
2.4 Teaching computational thinking to teachers.....	64
2.5 Models of School-based CPD	71
2.5.1 Lesson study as a form of CPD	75
2.6 Curriculum design in schools.....	78
2.7 Working with teachers or middle leaders to bring about change in schools.....	80
2.8 Summary	83
2.9 Research Aims.....	85
CHAPTER 3: Methodology and Methods	86
3.1 Introduction	86

3.2 Research Aims.....	86
3.3 Design-Based Research.....	87
3.4 Theoretical perspective	91
3.4.1 Pragmatism	92
3.5 Research Design.....	93
3.5.2 Phase 2- Developing Computational Thinking within a one- form entry school.....	98
3.6 Sampling.....	103
3.7 Materials and Data Analysis	103
3.7.1 Semi-structured interviews	104
3.7.2 Publicly Available Documentation from The Watermill School.....	106
3.7.3 Observations and Field Notes.....	106
3.8 Data Analysis	107
3.9 Research Validity and Reliability	121
3.10 Ethical Considerations.....	123
3.10.1 Informed consent and privacy	124
3.10.2 Confidentiality, Anonymity and Data Protection	124
3.10.3 Non-maleficence.....	126
3.11 Summary	126

CHAPTER 4: Data Analysis and Findings.....	127
4.1 Phase 1.....	129
4.2 What parts of the curriculum are being taught, and for how long?	130
4.2.1 The Planned Curriculum.....	130
4.2.2 The Enacted Curriculum.....	131
4.2.3 Who is teaching computing?	134
4.2.4 When is computing taught?	135
4.2.5 How is computing taught?	136
4.2.6 Summary.....	138
4.3 What factors are perceived to influence the teaching of computing within the school? (Challenges)	138
4.3.1 Challenges related to resources	139
4.3.2 Challenges related to students	142
4.3.3 Challenges related to teachers	144
4.3.4 Summary.....	147
4.4 Discussion of RQ1	149
4.5 Phase 2.....	152
4.5.1 Working with a Change Agent	153
4.5.2 Curriculum change	155

4.5.3 Career Professional Development in Schools	157
4.5.4 Summary.....	159
4.6 Research Question 3.....	160
4.6.1 Rewriting the Curriculum.....	160
4.6.2 Language in the Revised Curriculum	164
4.6.3 Teaching a Lesson	166
4.6.4 Continuing Professional Development of The Change Agent	170
4.6.5 Initial Continuing Professional Development for the Staff	172
4.6.6 Summary.....	178
4.7 Research Question 4.....	178
4.7.1 The Curriculum.....	179
4.7.2 The Change Agent.....	183
4.7.2 The Other Participants.....	189
4.7.4 Summary.....	191
CHAPTER 5: Discussion.....	192
5.1 Introduction	192
5.2 What challenges do primary schools face in teaching the computing curriculum?	192

5.2.1 The Curriculum.....	192
5.2.2 Challenges relating to resources	195
5.2.3 Challenges relating to students	197
5.2.4 Challenges relating to teachers	198
5.2.5 Language	201
5.2.6 Summary.....	202
5.3. How can we support schools in overcoming these challenges?	203
5.3.1 Curriculum redesign	203
5.3.2 Model lessons as a form of professional development.....	206
5.3.2 Working with a co-designer	207
5.3.3 What were teachers' experiences of and outcomes from the intervention? CPD?.....	210
5.3.4 Summary.....	212
5.4 Contributions of this thesis.....	212
5.4.1 Middle Leaders as Change Agents in Primary Computing	213
5.4.2 Supporting Novice Teachers to Teach Primary Computing	214
5.4.3 Implications for Future Curriculum Reform in Schools....	215
5.5 Contributions to my professional practice	217

5.5.1 Developing as a researcher	217
5.5.2 Developing as a practitioner	218
5.5.3 Summary.....	219
5.6 Limitations and Recommendations for Further Research.....	220
5.7 Conclusions	224
REFERENCES	225
APPENDICES	238
Appendix 1: Semi-structured interview questions for the head teacher and computing co-ordinator, adapted from Israel et al (2015)	238
Appendix 2: Class teacher interviews	239
Appendix 3: Lesson plan for teaching micro:bits	242
Appendix 4: CPD Lesson Plan.....	245
Appendix 5: Ethics documents.....	249
Participant teacher's consent form	249
Participant Teacher's information sheet.....	251
Parents' Information Sheet	254
Participants GDPR Information Sheet.....	257

Table of Tables

Table 1.1 Teachers at The Watermill School from 2017 to 2019	22
Table 3.1 Data Collected During Phase 1	95
Table 3.2 Data Collected during Phase 2	100
Table 3.3 Semi-structured interviews conducted in Phase 1	104
Table 3.4 Deductive Coding scheme (Scheme C) for document analysis of planning and curriculum documents based on The National Curriculum Computing Programme of Study (DofE, 2013b).	113
Table 3.5 Inductive Coding Scheme for Phases 1 and 2 (Scheme A)	116
Table 3.6 Deductive coding scheme (Scheme B) based on Sentance and Csizmadia (2017) with inductive codes (highlighted) developed from the data.	117
Table 4.1 Instances of the National Curriculum Objectives in The Watermill School's planned curriculum.	131
Table 4.2 Physical Computing Challenges in The Watermill School	139
Table 4.3 Student challenges identified in the literature	142

Table 4.4 Challenges related to teachers at The Watermill School	144
Table 4.5 Instances of the National Curriculum Objectives in the Watermill School's updated planned curriculum compared to the original curriculum	162
Table 4.6 Vocabulary to be taught per year group	165
Table 4.7 Computing curriculum objectives from teachers' medium-term plans. Computer science objectives are shaded.	180
Table 4.8 Challenges related to teachers at The Watermill School by order of occurrence	184

Table of Figures

Figure 2.1 Kennedy's Spectrum of CPD (Williams et al., 2022, p. 29)	72
Figure 2.2 Theoretical Model of the links between teachers' beliefs about learning and teaching and their participation in CPD (de Vries et al., 2013).	74
Figure 2.3 Teacher Curriculum Competence (TCC) (Tran and O'Connor, 2024, p. 8).	78
Figure 4.1 Data collected at The Watermill School from September 2017 to October 2019	128
Figure 4.2 Block overview for Year 3 at The Watermill School	132
Figure 4.3 The BBC micro:bit (micro:bit.org)	167
Figure 4.4 Illustrative slides from the taught lesson plan on An Introduction to micro:bits.	168
Figure 4.5 Slides used to demonstrate the concepts of decomposition and pattern recognition in mathematics.	175
Figure 4.6 An example block curriculum plan from The Watermill School for Year 4	177
Figure 4.7An annotated curriculum plan for Year 3	189

Glossary of Terms

Computational Thinking (CT): A set of problem-solving processes involved in formulating problems and expressing solutions in ways that a computer, or a human, can effectively carry out. Includes skills such as decomposition, pattern recognition, abstraction and algorithmic thinking.

Continuing Professional Development (CPD): Ongoing professional learning undertaken by teachers and other education practitioners to deepen their knowledge, strengthen their skills, and enhance their practice throughout their careers. CPD may involve structured, formal opportunities, such as accredited courses, workshops or conferences, as well as self-directed learning, including reading, reflection, collaboration and engagement with professional communities.

Core and foundation subjects. Maintained schools in England are legally required to follow the statutory national curriculum set out in programmes of study, based on key stages and subject content for those subjects that should be taught to all pupils. All schools must publish their school curriculum by subject and academic year online (DfE, 2013b, p. 5). Under the English National Curriculum, English, mathematics, and science are considered ‘core’ subjects and are taught more often during the school week. English and mathematics are taught for an hour a day, while science is taught for two hours a week. This can

put time pressure on the rest of the school week when the other ‘foundation’ subjects are taught.

The foundation subjects expected to be taught in English primary schools at Key Stages 1 and 2 include Art, Computing, Design and Technology, Geography, History, Languages, Music, Physical Education (PE), and Religious Education (RE). Schools are also expected to provide teaching for Personal, Social, Health Education (PSHE).

Higher Level teaching Assistant (HLTA): A teaching assistant who has achieved an additional qualification and taken on wider responsibilities, often supporting teaching and learning by leading small groups, covering classes under supervision and delivering specific interventions.

Interventions: Children who are considered to be working below the expected level of progress (usually in English and mathematics) are often taken out of the main class teaching to participate in individual or small group ‘interventions.’ These are often teaching assistant (TA)-led, and while they can involve pre-teaching a topic, they are usually aimed at filling gaps in pupil learning. Due to time pressures, interventions often occur in the afternoon, which may result in pupils missing out on foundational subjects.

Key Stages and National Curricula: The education system in England is divided into different key stages.

Foundation Stage

Children start formal schooling in the autumn of the year when they turn five and enter the foundation stage, which follows the EYFS statutory framework for group and school-based providers (DfE, 2024a) which starts in Nursery, where children are judged against early learning goals based on seven strands of learning. Computing is not a part of the framework, although schools will start to introduce fundamental concepts, such as staying safe.

Key stage one (KS1)

Following Reception, children enter Key Stage 1, which covers Years 1 (ages 5-6) and 2 (ages 6-7).

Key Stage 2 (KS2)

Key Stage 2 covers children aged 7-11 and school years 3-6. Lower Key Stage 2 (LKS2) covers Years 3 and 4, and Upper Key Stage 2 (UKS2) covers Years 5 and 6.

The national curriculum covers both KS1 and KS2. Once children leave primary school and move into secondary school, they enter Key Stage 3 (Years 7-9; ages 11-13) and then Key Stage 4 (Years 10-11; ages 13-16).

Most children progress through the year groups/ key stages according to age. Under the English education system, there is no tradition of keeping children behind (or advancing them) through the year groups. The only exception is when children are born prematurely, in which case they may occasionally delay starting reception.

Newly Qualified Teacher (NQT): The first year post qualification, which should be an induction year that enables them to be legally employed as a teacher in a maintained school (DofE, 2018). The induction period lasted a year for full-time teachers, with each teacher having only one chance to complete it. If a teacher failed their induction year, they could not repeat it or legally teach in a state-maintained school. All NQTs were entitled to additional school support and PPA time to help them complete their induction period. The term Early Career Teacher (ECT) replaced that of NQT in September 2021 and now covers the first two years of teaching (DfE, 2023).

Planning, Preparation and Assessment Time (PPA): All teachers are allocated 10% of their timetable as non-contact planning, preparation and assessment time. This usually equates to half a day a week and is usually taken in the afternoon. A qualified teacher should cover PPA time, but in many schools, it is covered by HLTAs or sports providers. Some schools use specialist teachers to cover PPA time, teaching subjects such as music.

Pre-service Teacher: A trainee teacher who is undertaking initial teacher education but has not yet qualified or achieved Qualified Teacher Status (QTS). The term ‘student teacher’ can also be used.

Recently Qualified Teacher (RQT): This refers to the second year following qualification. This had no official standing in English schools, and teachers were not entitled to additional time or support. Since 2021, the

induction process has changed to two years, with teachers referred to as Early Career Teachers (ECTs) during this time (DfE, 2023).

SATS The UK Statutory Assessment Tests are national tests that children take twice during primary school. As a baseline assessment in Reception (ages 4-5), which assesses children's starting points in language, communication, literacy and mathematics (Standards and Testing Agency, 2024b) and again at the end of Key Stage 2 (KS2) in Year 6 (ages 10-11). These standardised tests are known as 'End of Key Stage Tests and Assessments', but most people know them as SATs.

Teachers also assess the Key Stage 1 SATs. There are papers available, but they are relatively informal, not sent off for external marking, and teachers have flexibility over when they are sat. Children are tested in maths (arithmetic) and English (reading, spelling, punctuation, and grammar). End of KS1 national curriculum tests and teacher assessments have been non-statutory since 2023-24, but the government encourages schools to use them still (Standards and Testing Agency, 2024a).

Key stage 2 SATs are much more formal. They are taken at set dates and times and sent away to be marked. Children will take exams in English reading comprehension, a combined grammar, punctuation and spelling test, mathematical reasoning, and arithmetic. Teacher assessment is used to assess children's writing skills.

KS2 SATs results are published, and schools are judged on whether sufficient progress has been made between the starting Reception and the end of KS2. Schools and parents will receive:

- Raw score –the actual number of marks they received in their SATs
- Scaled score – a conversion score that allows results to be compared year-on-year
- Expected standard – whether or not they have achieved the national standard

Schools are compared to a National Standard, the local authority, and schools with a similar socio-economic profile.

Pupils are tracked based on their attainment and progress scores from entering school, with emphasis on whether they are meeting, below, or exceeding national expectations in key areas (writing, grammar, punctuation and spelling, reading, and mathematics). Pupils are also tracked to ensure that they have made sufficient progress between the key stages. Poor scores are likely to trigger earlier Ofsted inspections and can also influence whether parents choose to send their children to a particular school.

Trends in International Mathematics and Science Study (TIMSS): A large-scale international assessment carried out every four years, measuring pupils' achievement in mathematics and science across participating countries. TIMSS also collects contextual data on teaching practices, curriculum, and school environments.

CHAPTER 1: Introduction

1.1 Introduction

As a practitioner-researcher, I will begin this chapter by describing my professional background and the origins of my interest in researching the computing curriculum in primary schools. I will explore the history of computing teaching in the English education system, before focusing on Computational Thinking (CT) and discussing how the role of the subject leader in English Schools influences its teaching. I will then introduce the context of my study by describing the one-form entry school I worked with, followed by an overview of the staff within the school. The chapter concludes with an overview of the thesis' goals and a description of its structure.

1.2 Overview/ Personal Context

In 2015, having worked as a classroom teacher and subject specialist in a variety of English Primary Schools (or elementary schools, ages 5-11) for 10 years, I began working in Higher Education as a University Lecturer teaching Primary Education subject knowledge and pedagogy with a particular focus on mathematics and computing, to pre-service teachers. This coincided with the introduction of a new English National Curriculum (DfE, 2013b) that led to sweeping changes across all subject areas.

One of the most significant differences in the curriculum was the change from teaching Information Communications Technology (ICT) to Computing. Conversations with pre-service primary education students

returning from Teaching Placements and observations of what was happening in school led to a growing awareness that the new computing curriculum was not being taught to its full potential or at all in some instances. Anecdotal evidence from students suggested that the subject was often overlooked or deliberately assigned to them as a teaching tool; they also mentioned not having sufficient resources to teach the subject when they were in school.

These conversations, coupled with my personal interest in the subject, led me to want to understand why this was happening and to change school practices so that computing could be taught to its full potential. This thesis constitutes my doctoral research in this area. Using a design-based research approach, I aim to explore teachers' experiences in teaching the computing curriculum in a one-form entry primary school and examine strategies to enhance teachers' subject knowledge and engagement with the new curriculum.

1.3 Computing in the English Education System

The teaching of Information and Communication Technology (ICT) was made compulsory for all pupils from 5 to 16 in English Maintained Schools since the Education Reform Act of 1988 (Ofsted, 2011). The primary focus of the 1999 National Curriculum was on pupils using ICT tools and information sources to support their work in other subjects, including developing research skills and presenting information in a way that suited an audience (Department for Education and Employment, 1999). There were

some elements of programming or coding, with pupils in Key Stage 1 being taught to how to ‘plan and give instructions to make things’ (p. 16) often using a floor robot or turtle and pupils in Key Stage 2 taught to ‘how to create, test, improve and refine sequences of instructions to make things happen and to monitor events and respond to them’ (p. 18) usually involving data loggers. Ofsted’s (2011) report on ICT in schools from 2008-2011 found that the teaching of ICT was good or outstanding in nearly two-thirds of primary schools visited, with many teachers and teaching assistants confident to support pupils effectively. However, in Key Stage 2, pupils in the majority of schools visited had ‘insufficient opportunities to develop their understanding and use of programming, and data logging and handling’ (Ofsted, 2011, p. 9).

The picture in secondary education (age 11-18) was less positive, with teaching and learning judged as no better than satisfactory in just over half of schools inspected between 2008-2011, with achievement judged as good or outstanding in only 27 out of the 74 secondary schools inspected (Ofsted, 2011). The curriculum focused firmly on using computers, but did not emphasise how they worked or their underlying principles (Computing at School Working Group, 2010). Ofsted’s (2011) report claimed that the ICT curriculum at Key Stage 4 was not adequately preparing pupils for higher-level academic courses, with the repetition of undemanding activities already covered in Key Stage 3, alongside a lack of challenge for pupils, meaning that fewer pupils were choosing to study ICT at GCSE level. This was argued to

lead to disenchantment with studying ICT and further declines in the numbers studying ICT or computing at A-Level and at University, with a 60% decline in applications to read Computer Science in UK universities from 2000 to 2009 (Computing at School Working Group, 2010). Concerns expressed from varied perspectives, including schools, business and industry, and universities, led to a broad review of the state of computing in schools by the Royal Society (Furber, 2012), with proposals submitted to the Government regarding the development of a new curriculum for computing. The then Education Secretary, Michael Gove, announced in 2012 that the ‘harmful’ ICT curriculum would be scrapped, to make way for a new computing curriculum (Gove, 2012).

This new primary national curriculum for computing (DofE, 2013b), introduced alongside large-scale changes to all subjects, led to serious challenges for all schools, especially primary schools, with the official change from the national curriculum foundation subject of Information and Communication Technology (ICT) to Computing across all four key stages. This required an Order to replace ICT with computing under Sections 84 and 85 of the Education Act 2002 (DofE, 2013a).

The change to the curriculum led to significant adjustments to what should be taught in primary schools, with the curriculum divided into:

‘(1) computer science (the rigorous academic discipline that covers algorithms, data structures, programming); (2) digital skills (the general ability

to use computers confidently, effectively and safely); and (3) information technology (the design and application of digital systems to meet user needs for particular purposes) (DfE, 2013a, p. 2)’.

Although it is not a mandatory GCSE subject, computing is compulsory until age 16 (Ofsted, 2022). The curriculum is non-prescriptive, with no prescribed coding languages, software or hardware to use (The Royal Society, 2017), which has, in itself, caused challenges for schools unsure of the best approaches or equipment to use.

The most significant change to the curriculum can be seen through the introduction of computer science, described by Computing At Schools as:

‘an academic discipline with its own body of knowledge that can equip pupils to become independent learners, evaluators and designers of new technologies’ (Csizmadia et al., 2015).

Computer science accounts for half of the aims of the 2014 National Curriculum for computing (DofE, 2013b, p. 2) and is explicitly referenced in the subject content in both Key Stages 1 (ages 5-7) and 2 (ages 7-11), where at Key Stage 1, pupils should be taught:

- ‘Understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions
- create and debug simple programs

- use logical reasoning to predict the behaviour of simple programs.’

With Key Stage 2 adding in:

- ‘Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- use sequence, selection, and repetition in programs; work with variables and various forms of input and output

use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs (DfE, 2013c, p. 2).’

The introduction of computer science to the curriculum has led to noteworthy challenges for teachers and schools across all key stages, particularly through the introduction of coding/ programming to the curriculum. The majority of the secondary school teachers teaching ICT in schools come from non-computing backgrounds and have had to retrain (Peyton-Jones et al., 2013). A 2019 Royal Society policy briefing found that secondary schools struggled to recruit teachers, with the number of entrants to teacher training 1,012 fewer than required to keep up with the demand. In addition to this, fewer schools are offering computing courses, with fewer options available (The Royal Society, 2019).

The changes to the curriculum are even more significant within primary, as Peyton-Jones et al. (2013, p. 10) state:

‘Launching a new school subject from a near-zero base is no easy matter. Will the universities really be willing to offer professional development courses for free? Will teachers be willing to learn? Will head teachers be willing to give them time off to learn? In primary schools, there are few specialist ICT teachers; will primary schools really be able to teach Computer Science?’

As Larke (2019) notes, no additional funds were provided to support teacher training or to purchase new infrastructure. This, alongside the absence of any official computing guidance, meant that primary schools and their teachers, who were mainly subject generalists, were left to devise their own curriculum.

The standard teaching model in a one-form primary school would involve the class teacher teaching the computing curriculum unless the school employs a specialist teacher; this is the case for most subjects in the English national curriculum, where primary school teachers would typically expect to teach all subjects.

The majority of teachers working within primary schools, therefore, have little understanding of computer science and no experience in programming, which leads to a change in the subject knowledge and,

therefore, the technological pedagogical content knowledge (TPACK) they require. Teachers who have no formal education in computing often struggle to show enthusiasm for the subject and do not have enough understanding to explain how computers work (Wilson & Moffat, 2010).

Computing is not part of the annual Statutory Testing (SATs) of children in English schools at Key Stage 1 (Year 2) or Key Stage 2 (Year 6), which are focused on English and mathematics and have a significant impact on how schools are judged. In addition to this, whilst the most recent Ofsted education inspection framework (EIF) for primary schools emphasises the ‘connectedness of the curriculum,’ it focuses on reading and mathematics, with deep dives into other subjects, looking mainly at those that are being taught, during the inspection (Ofsted, 2019, p. 3).

Typically, in English schools, subjects such as English and mathematics are taught in the morning when children are considered to be more focused and, therefore, better engaged in learning. As teaching English and maths in the morning has been shown to increase pupils’ grade scores (Pope, 2016), and these subjects are the ones on which schools are currently judged, they are seen as having a higher status in some schools. They will often be prioritised over ‘afternoon’ subjects, including the humanities, religious education and computing. Children may even be taken out of the ‘afternoon’ subjects for intervention (booster sessions) in English and mathematics. The English National Curriculum (DfE, 2013b) consists of

eleven compulsory subjects that schools must teach to all pupils, as well as providing RE and relationship and health education, all of which need to be fitted into a relatively short school week. Consequently, schools often struggle to allocate sufficient time to all subjects (Clark-Wilson et al., 2019), which can mean that subjects that teachers do not feel confident teaching may get marginalised or delegated.

Studies examining computing education three to five years after the curriculum's introduction (Larke, 2019; Sentance & Csizmadia, 2017; The Royal Society, 2017) indicate patchy uptake. Many teachers focus on the parts of the curriculum that they already know or feel confident teaching, a pattern particularly evident in primary schools, where weekly time pressures are significant (Israel et al., 2015). This has not changed in subsequent years. This means that, if computing is taught at all, Information Technology and Digital Skills are often given precedence, whilst Computer Science and, therefore, Computational Thinking (CT) are left until the end of the year or do not get taught at all. The picture seems even worse in secondary education, with the SCARI report into subject choice and attitudes to computing, 10 years after the introduction of the curriculum (Kemp et al., 2024) finding that only 28% of young people were choosing to study a digital qualification at Key Stage 4 in 2023 due to perceived relative difficulty and not enjoying the subject, and teachers still feeling unprepared to teach the curriculum.

1.3.1 Computational Thinking in Primary Education

Even though Computational Thinking (CT) is a common term within the computer science education community, amongst many teachers, administrators and teacher educators, there is confusion about what it entails (Yadav et al., 2017).

Computational thinking, initially defined by Wing (2006) in a seminal article as an approach to designing systems, solving problems and understanding human behaviour that draws on concepts key to computing, provides a framework for studying computer science with wide applications beyond computing itself (Csizmadia et al., 2015). Wing (2006, p. 33) argued that computational thinking is a ‘universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use’ and, therefore, should be included in all children’s education as a key part of school learning. Barr and Stephenson (2011) note that whilst children already learn how to think and problem-solve, computational thinking can help students understand these processes as algorithmic and identify where actual computation and manipulation of data with a computer may fit in.

There is a lack of consensus around the definition and scope of CT (Kite et al., 2021) although Kite et al. (2021) suggest that research can be divided into studies that focus on code-centric *skills* (such as debugging) and those that use *interdisciplinary practices*. This is similar to the division of CT into *key concepts* (Logic, Algorithm, Decomposition, Patterns, Abstraction

and Evaluation) and *approaches* (Tinkering, Creating, Debugging, Persevering and Collaborating) commonly used in UK schools (Barefoot Computing, 2017). Brennan and Resnick's (2012, p. 1) often cited framework identifies three key dimensions of CT, which they suggest can be used to assess the learning of CT concepts in the classroom:

'Computational concepts (the concepts designers engage with as they program, such as iteration, parallelism, etc.), *computational practices* (the practices designers develop as they engage with the concepts, such as debugging projects or remixing others' work), and *computational perspectives* (the perspectives designers form about the world around them and about themselves)'.

Computational thinking does not require knowledge of computer science and should be seen as a way that humans, not computers, think (Wing, 2006). Instead, it provides 'a set of skills and analytical abilities that are crucial in the modern society pervaded with technology in every aspect of everyday life' (Towhidnejad et al., 2014, p. 1) and, therefore, is interwoven throughout numerous scientific and mathematical disciplines. This means that, whilst computer science was the original field and practice of computational thinking, it is not the only discipline in which the skills can be found or developed (Voogt, Fisser, et al., 2015). Robbins, Senseman and Pate (2011) agree with this, arguing for the importance of teaching computational thinking in context.

For this study, Shute, Sun, and Asbell-Clarke's (2017, p. 142) working definition of Computational Thinking (CT) will be used:

‘The conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts.’

This definition asserts that CT is primarily a way of thinking and acting that can be demonstrated across various contexts, and it does not require a computer. To understand what this looks like in schools, it is necessary to understand what ‘thinking like a computer scientist’ means and, therefore, the concepts and practices or approaches that are taught.

There are several concepts or elements of the thinking process that designers use as they program, or computer scientists engage in as they solve problems associated with CT. As the terms defined by Barefoot Computing (2017, 2021) are the most frequently used in UK primary schools and in Initial Teacher Education, these will be used (in bold below) alongside further definitions by Grover and Pea (2018) and linked to those being used by authors in the literature.

These include:

- **Logic** (and Logical Thinking) involves helping to establish and check facts, making predictions (Barefoot Computing, 2021) and analysing

situations to make a decision or reach a conclusion about them (Grover & Pea, 2018).

- **Evaluation** involves making judgements about solutions based on how well they solve a problem or goal (Barefoot Computing, 2021; Grover & Pea, 2018).
- **Algorithms** (and Algorithmic Thinking) are a precise sequence of instructions, rules or step-by-step instructions designed to meet an end goal, perform a task or solve a problem (Barefoot Computing, 2021; Grover & Pea, 2018). In K-12 education, the algorithms usually involve *sequencing*, including *conditional* checks (such as if-then-else patterns), *selection* and repetition involving *loops* (such as do-while, repeat or repeat until) (Grover & Pea, 2018).
- **Patterns** (and Pattern Recognition) involves spotting and using similarities to help make predictions and create rules which can then be used to solve problems (Barefoot Computing, 2021). In CT terms, pattern recognition can be used to recognise parts of problems that have previously been solved, so that they can be reused (Grover & Pea, 2018).
- **Abstraction** involves removing unnecessary detail by identifying what is important and removing unnecessary detail (Barefoot Computing, 2021). Grover and Pea (2018) consider this to be the foundation of computer science and, therefore, CT, providing a way to simplify and, therefore, generalise based on similarities and differences.

- **Decomposition** involves breaking a problem or system down into its parts (Barefoot Computing, 2018). It is not unique to CT but is a key part of computational problem-solving, as breaking a problem into smaller sub-problems makes it more manageable (Grover and Pea, 2018). It should be noted that while Barefoot Computing (2017) classify decomposition as a CT concept, Grover and Pea (2018) consider this to be an approach that computer scientists might use when engaged in CT problem-solving.

Several practices or approaches have been associated with Computational Thinking:

- **Creating** linked to planning, designing, making and evaluating artefacts (Barefoot Computing, 2021) a computer can execute, which is often considered to be the end goal of CT and problem-solving (Grover & Pea, 2018). These can include computational artefacts, such as games, stories, or simulations, that will eventually become a physical artefact. This can be linked to another approach identified by Barefoot Computing (2017): **Tinkering**, experimenting, and playing while changing things to see what happens.
- **Debugging** (and Testing) concerns discovering and fixing what is wrong with an algorithm or program (Barefoot Computing, 2021). It can be considered integral to any problem-solving and an essential part of evaluating a computational solution (Grover & Pea, 2018).

Persevering to solve a problem is often linked to this, with students encouraged to persist and demonstrate resilience (Barefoot Computing, 2021).

- **Collaborating** is often described as a common practice in computational problem-solving. It is often encouraged in K-12 classrooms through pair programming or requiring students to alternate between taking the lead on typing or reviewing code (Grover & Pea, 2018).

It can be challenging to introduce too many concepts into a project with young children, so researchers or teachers in schools may focus on a smaller number of key concepts rather than introducing too many at once. Progression through a curriculum should mean that concepts introduced to younger children are reinforced and built upon throughout a planned curriculum or teaching sequence.

As previously mentioned, whilst what should be taught is prescribed through the National Curriculum programmes of study (DfE, 2013c), how computing is taught is left to individual schools to decide. This is usually left to the school's computing co-ordinator to organise. The following section will, therefore, examine the role of subject leadership in English Primary Schools and the impact it can have on how subjects are taught.

1.3.2 The Role of Subject Leadership in English Primary Schools

Teachers in English Primary schools are expected to take on responsibility for a specific area of the curriculum; this might be linked to a teacher's primary degree, an interest or the 'necessity' of finding someone willing to take on a subject (Hammersley-Fletcher, 2002). Titles used to describe teachers with subject responsibility in English primary schools vary, with the role evolving over time to increasingly involve curriculum support, particularly with the advent of the National Curriculum, which places more emphasis on teacher subject responsibility and accountability (Hammersley-Fletcher, 2002). Busher, Harris, and Wise (2000, p.5) argue that in primary schools, the role is complex, as teachers have responsibility for subject leadership across Key Stages, but without 'the authority of responsibility or status.' The role varies in different schools but includes advocating for the subject (Busher et al., 2000), preparing for and discussing with Ofsted Inspectors (Ofsted, 2022), writing schemes of work and resource management (Hammersley-Fletcher, 2002) and working with colleagues to produce change and improvements to teaching and learning in response to changing policies and curriculums (Busher et al., 2000).

In small schools, teachers are more likely to lead multiple subjects, with rural and small schools often relying on informal rather than formal teacher leadership roles (Wiens & Beck, 2022). This can lead to time pressures with senior and middle managers, particularly in primary schools, who have

little time for the managerial aspects of their work (Bush, 2020), especially as primary subject leaders will usually be full-time teachers, teaching every subject.

Subject leaders in English primary schools can be viewed as teacher leaders, as identified by Nguyen, Harris, and Ng (2020), as a necessary requirement for schools to run effectively. Definitions of teacher leadership vary; Wenner and Campbell (2017) define teacher leadership as teachers who assume leadership responsibilities outside the classroom. In contrast, Nguyen et al. (2020, p. 67) take a broader view of leadership, arguing that teaching and leadership should be integrated, both within and beyond the classroom. Teacher leaders are seen as influencers, leading to a change with specific teachers acting as ‘key change agents and sources of innovation.’ Harris and Jones (2019) argue that teachers should be viewed as co-constructors of educational change and key contributors to policy change, with teacher agency, collaboration, and leadership being crucial to improving learner outcomes.

Wiens and Beck (2022) suggest that the key roles of teacher leaders are as resource providers and instructional supporters for their colleagues; however, they note that there is little evidence that teacher leaders have the skills to fulfil these roles compared to their peers. Transformational leadership, however, positively impacts the commitment of teachers to an organisation

and increases the willingness of teachers to ‘exert effort and to change their classroom practices and attitudes towards change’ (Wang et al., 2020, p. 13).

An interested and passionate subject leader can have a significant impact on how a subject is taught within a school; this is particularly true for computing, which can cause teacher anxiety. The following section of the thesis introduces the research context, including the staff and the computing coordinator.

1.4 Context

This next section of the thesis aims to provide a rich description of the research context by describing the one-form entry school in which it is situated and the staff within the school.

1.4.1 The School

The Watermill School¹ is a mixed one-form-entry school, one of two in a semi-rural small town in Northwest England with a population of around 5,000. The school is slightly smaller than the average primary school size of 276 pupils (DfE, 2024b), with 30 pupils per year from Reception to Year 6 (ages 4-11). The town has a secondary school that also serves the neighbouring town. Most children in the area attend the local secondary

¹ The school and all staff have been given pseudonyms to preserve anonymity.

school, which has a good reputation and achieves above-average results at GCSE and A-level.

The school is situated in an area of mixed housing², which is reflected in its student body; there is a slightly lower percentage of children receiving free school meals than the national average, an average percentage with a statement for special educational needs, and a lower-than-average percentage of bilingual children. The school has a higher-than-average rate of persistent absence, 23.7%, compared to a national average of 18.7% (DofE, 2025), which can impact attainment.

The current head, Sandra, has been at the school since 2013. Before her arrival, the school did not have a robust reputation; during an Ofsted (Office for Standards in Education, Children's Services and Skills) inspection in 2014, however, the school was graded as good across all areas; the report noted the change in leadership and that improvements since the previous inspection had been rapid. The report specifically stated, 'The strong leadership and drive of the deputy head teacher and the new head teacher have significantly improved teaching so that pupils' achievement is good' (Ofsted, 2014). The school underwent a new inspection in 2019 during the study and was again graded as 'good'. This is despite a significant dip in the school's Key Stage 2 statutory

² Mixed-Economy housing refers to residential developments that combine different types of housing occupancy and affordability within the same community. These aim to address inequalities in access to housing and community participation (The Oxford Review. (n.d.)

assessments for 2018, where only 32% of children met the expected standards across all subjects, compared to 64% for the national average and the local authority average, with no children achieving the higher standard, compared to 9% nationally (DfE, 2018). The Ofsted report recognised the nature of the children attending the school and the improvements the school was making to monitoring.

Sandra is a strong leader who is keen to engage in educational research. She is not afraid to make decisions that could be seen as controversial, for example, stopping written feedback in books where the evidence shows that it does not affect children's learning. The school remains up to date with current thinking, and there is a strong emphasis on all staff taking on leadership roles.

The school divides its school year into four equal teaching blocks. These cross over school holidays and are designed to compensate for the fact that school terms are often unequal in length. Classes follow a timetable and topic for a block. Teaching is mainly thematic, with most teaching planned around a central theme or topic.

From the school website, when I first viewed it in 2017, it was evident that computing had not been a high priority for the school up to that point. There was little mention of computing in curriculum plans. While all classes had blogs updated weekly, many posts were dated or updated by the computing coordinator. E-safety, however, was seen as a priority and had a clear presence on the webpage with links to information for parents. The school had recently invested a small sum of money in a set of Lego WeDo (LEGO Education, n.d.)

resources for computing and, during the project, invested in a set of micro:bits to use with the children.

1.4.2 The Staff

The following teachers taught at the school throughout the research period from September 2017 to November 2019. The school had limited teacher movement during the period (one teacher retired, and a Newly Qualified Teacher (NQT) was appointed). All staff have been given pseudonyms. A summary of the key points is presented in Table 1.1 below.

Table 1.1*Teachers at The Watermill School from 2017 to 2019*

Teacher ³	Roll	Year Groups ⁴ Taught ⁵	Years of experience	Computing experience/ training/ interest (self-reported during interviews)
Alex	Computing co-ordinator Classroom teacher PPA teacher (2018-2019)	2017-2018- Year 2 2018-2019 – teaching across year groups 2019-2020 – Year 2	5+ years	Interested in computing. Given the role of computing co-ordinator as ‘most competent at IT’ Confident teaching computing. Mainly taught the post-2013 curriculum but trained at university under the previous curriculum. Helped to organise Continuing Professional Development (CPD). Has attended computing CPD.
Caroline	Deputy head Mathematics co-ordinator	2017-2018 – Year 6 2018-2019 – Year 1 2019-2020 – Reception	10+ years	Avoided computing teaching when teaching year 6. Trained at university under the previous curriculum. Did not attend CPD.
Chloe	Class teacher	2017-2018 (Not present)	Recently qualified	Less confident teaching computing than other subjects.

³ All names are pseudonyms⁴ In English schools, children start Reception in the academic year (Sept-Aug) when they turn 5. Reception Age 4-5; Year 1 Age 5-6; Year 2 Age 6-7; Year 3 Age 7-8; Year 4 Age 8-9; Year 5 Age 9-10; Year 6 Age 10-11.⁵ Whilst some teachers will always teach the same year group, it is common for teachers to move around.

		2018-2019 – Year 3 2019-2020 - Year 3	teacher at the start of 2018- 2019.	Recent computing training at university as part of a 1-year PGCE Attended CPD
Hannah	PPA teacher Class teacher	2017-2018 - PPA teacher 2018-2019 – Year 4 2019-2020 – Year 1	10+ years	Was the ICT coordinator pre-2014 under the old curriculum Confident teaching Information Communication Technology (ICT). Less confident teaching computing.
James	Class teacher	Year 5 throughout the project	5+ years	Originally trained as a secondary English teacher, so no formal computing training, but learned ICT at school. He is enthusiastic about technology but less confident in his knowledge of computing. Agreed to be observed teaching computing. Attended CPD.
Matt	Class teacher	2017-2018 (Not present) 2018-2019 – Year 2 2019-2020 – Year 4	Newly qualified teacher at the start of 2018- 2019	Three-year education degree with computing covered throughout Studied ICT in high school Strong subject knowledge and confidence in teaching Was supporting Chloe with planning computing in 2019- 2020 Attended CPD
Rose	Class teacher	2017-2018 – Year 3	20+ years	Taught under the pre-2014 curriculum

		2018-2019 – Retired		Confident teaching ICT (word processing), not confident teaching ‘coding’. Agreed to be observed teaching computing.
Sally	Class teacher	2017-2018 – not present 2018-2019 – Year 6 2019-2020 – Year 6	10+ years	Confident in teaching ICT, less so in computing due to lack of experience. Attended CPD
Sandra	Headteacher	Non-teaching management position	10+ years	Good knowledge of the computing curriculum, but has not taught it.

Alex

Alex is the school's computing coordinator and has been in this role for seven years, having started as a newly qualified teacher (NQT) at the beginning of the project. He took over the role of computing co-ordinator four years ago when the previous co-ordinator left. Alex is in his early thirties and has a young family. He is interested in computers/ computational thinking and is especially passionate about e-safety. Alex was keen to get involved in the project, has been happy to trial new approaches, and has invested time and money in new resources and training, as well as updating key documents such as the school's computing curriculum. He has attended CPD in computing in the past.

At the start of the study (2017-18), Alex was teaching Year 2 (ages 6-7). However, for the main year of data collection (2018-2019), Alex took on a part-time role with a charity and went part-time at the Watermill school. He was working three days a week, covering Planning, Preparation and Assessment (PPA) time. Most of the teachers took advantage of this to have Alex teach their computing lessons, especially those focusing on computational thinking. However, the funding for the charity work was only for one year, and Alex returned to teaching full-time for the 2019-2020 school year, again teaching Year 2.

Caroline

Caroline is an experienced teacher in her forties, the school's deputy head and mathematics lead. Despite initially agreeing to an interview, the interview was repeatedly delayed, and she did not attend the CPD. In the first year of the study, when she was a Year 6 (ages 10-11) teacher, she delegated

all of her computing teaching to the teacher covering her Planning, Preparation and Assessment (PPA) sessions.

Chloe

Chloe is a Recently Qualified Teacher (RQT) (teaching for less than two years) in her mid-twenties. Despite her recent university training, she feels less confident teaching computing than other subjects.

Hannah

Hannah is an experienced teacher in her late forties. She had previously been an ICT coordinator, so she was confident teaching the digital literacy parts of the curriculum. However, she feels less confident about teaching computational thinking and using newer technologies and software. She was teaching PPA for the first year of the study and was involved in model computing lessons with the children; for the remainder of the study, she was responsible for her own class.

James

James is an experienced teacher in his late thirties. He originally trained as a secondary school English teacher, so he has had no formal training in computing; however, he is enthusiastic about technology. He is willing to 'have a go' when teaching computing, though he finds some of the terminology daunting. He was happy to be observed teaching computing.

Matt

Matt is a Newly Qualified Teacher (NQT) in his twenties. He was teaching Year 2 (ages 6-7) for the 2018-2019 academic 2017-2018 and was

appointed on a temporary contract. As an NQT, he received extra PPA time, which Alex covered for the year, meaning Alex taught Year 2 for a full day. Matt was keen on teaching computing and was frustrated by the time allocated to training for it. Whilst he considered his subject knowledge of the National Curriculum to be not that strong, he is excited by the prospect of teaching computing. However, as Alex taught his class for a full day, he tended to focus on teaching the computing (and especially the computational thinking sessions) rather than Matt.

At the start of the 2019-2020 academic year, when the CPD took place, Matt was teaching full-time in Year 4 (ages 8-9) as an RQT. He supported Chloe with the planning and teaching of computing in Year 3 (ages 7-8).

Rose

Rose is an experienced teacher in her sixties, nearing retirement. She was accustomed to teaching the old ICT curriculum and felt more confident teaching 'word processing' than the new one. She was happy to be observed teaching computing and completed a written questionnaire. She retired at the end of the first year of the study.

Sally

Sally is an experienced teacher in her forties. She joined the school in 2018 and was not interviewed as part of the study; she was an active participant in the CPD and was keen to take up the offer of additional support.

Sandra

Sandra has been the head teacher at Watermill School since 2013 and has implemented significant changes to the school's staffing and ethos. She was keen to involve the school in research and is not afraid to try new approaches if they have been proven to improve children's learning.

The context of the study is crucial for understanding its location within the research landscape. Providing a rich picture allows the unique situation of the research to be seen, but also allows generalisations to be made to similar contexts (Thompson & Hall, 2017). The following section will introduce the research aims within the school context.

1.5 Thesis Goals

This research aims to answer the overarching research question: *How can the teaching of computing for primary school children be improved in an English Primary school?* In this thesis, I will explore the impact of introducing computing to the National Curriculum (DfE, 2013b) within a one-form entry primary school. Beginning three years after the introduction of the curriculum, I will start by describing any challenges and barriers to teaching computing, before exploring possible solutions to improve computing teaching, leading to better outcomes for both staff and students. In this thesis, I aim to argue that a lack of support for teachers in schools exacerbated challenges and that involving middle leaders in situated continuing professional development (CPD) is essential for any successful curriculum reform.

1.6 Layout of thesis

This chapter identifies my position as a practitioner-researcher and provides an overview of the background and context of computing in the English primary school curriculum. It began with an overview of the history of computing in the National Curriculum, followed by an examination of computational thinking in further detail and a discussion of how subject leadership in English schools influences the teaching of subjects. It finally introduced the context of this current study through a description of the school and its teachers.

This thesis has four further chapters. Chapter 2 (Literature Review) details the background to the study. It begins with a discussion of Computational Thinking (CT) in education, examining why CT should be taught in schools, when it should be introduced, and how it is being taught. Secondly, the chapter explores teachers' perceptions of teaching CT, before examining factors that may prevent the teaching of CT in schools. Thirdly, the chapter reviews research into models of enhancing teachers' CT subject knowledge through continuing professional development (CPD) and curriculum design, before finally discussing how working with middle leaders in schools can help to facilitate this. The chapter concludes by outlining the research aims and questions, based on the identified gaps in the literature review.

Chapter 3 (Methodology) describes the methodology and methods used to generate the data. It details the methodological approach, design-based research (DBR) and describes the pragmatic aim of the research. The chapter also describes the two phases of the research, the data corpus (Braun &

Clarke, 2006) generated through a range of materials, and how thematic analysis was carried out. The chapter concludes with a consideration of the ethics of the current study, including research validity and reliability.

Chapter 4 (Data Analysis and Findings) focuses on the findings and data analysis of the two phases of the study. Phase 1 examines the picture of computing in a one-form entry primary school three years after the introduction of the 2014 curriculum to find out how the planned curriculum has been developed and enacted in the school and how computing is being taught, before moving on to challenges that influence the teaching of computing within the school to answer RQ1. Phase 2 aims to answer RQ2-4 by working with a change agent to revise the curriculum and develop continuing professional development (CPD) for both the change agent and other staff members in the school.

In Chapter 5 (Discussion), I summarise the extent to which the research has achieved its aims by discussing the findings in relation to existing research and indicate the contributions made to knowledge and professional practice in this research. Chapter 5 also discusses the limitations of this research, identifies areas for future research, and suggests potential improvements.

CHAPTER 2: Literature Review

2.1 Introduction

Wing (2006) stated that Computational Thinking (CT) is such a fundamental skill that it should be added to reading, writing and arithmetic as part of every child's analytical ability. This literature review will examine the benefits and challenges of teaching CT to students, with a focus on those in Primary or Elementary Education (ages 4-11).

The first section of this review will consider the arguments about whether we should teach CT to primary pupils. It will then discuss when and how CT should be taught before addressing teachers' perceptions of teaching it.

The second section of this review will explore the challenges of integrating CT into the primary curriculum and examine factors that can impact whether CT is taught. The final section will consider what we can do to improve the teaching of CT in schools, exploring how CT has previously been taught to teachers to enhance their subject knowledge and perceptions of the subject. Finally, it will explore how computing is managed by middle leaders in schools and the impact that this can have on how computing is taught.

2.2 Computational Thinking (CT) in Education

As discussed in Chapter 1, Computational Thinking was introduced to the English National Curriculum as part of the Computing Programme of Study in 2014 (DfE, 2013c). To understand the impact of CT on computing teaching, it is essential to discuss why it is considered an important concept to

teach, when it should be introduced, and teachers' responses to its inclusion in the curriculum.

2.2.1 Why should Computational Thinking be taught?

Several authors have described computational thinking (e.g. Falloon, 2024; Sentance & Waite, 2018) as a fundamental skill and an essential part of the curriculum. Wing's (2010) seminal article argues that CT has benefits both through its application in other disciplines, such as mathematics or medicine, and in everyday life. Guzdial (2015) argues, however, that while there is evidence for the former claim, there is less support for the idea that CT can be helpful in daily life. Therefore, this section will explore the importance of computing and CT, examining its potential applications in daily life. It will also consider why it must be included in the curriculum and taught explicitly in schools.

In his book on computing education, Guzdial (2015, p. 22) summarised the key reasons why he believes everyone should learn computing (including increased computational literacy, productivity, and broadening participation) and reviewed research in these areas. He argues that the aim of a computationally literate society is 'to use computing as a form of expression and a way to think about domains other than computing,' with computing allowing people in many fields, such as medicine, engineering or mathematics, to do things that they could not do without computing. He argues that being computer-literate enables end-users of computing to save time and think about their domains in new ways. Guzdial (2015) goes on to say that teaching computing to all has two benefits to productivity: the first being that students can do more with computing, and the second being that students who

understand too little about computing are likely to be less productive as they make uninformed choices about the tools to use or ways to use them.

Sentance and Waite (2018), arguing for the importance of mandatory computing lessons (specifically including CT) in the English National Curriculum, agree with the importance of CT skills for all, arguing that these skills not only help to prepare students for the types of problem-solving they will need to be doing in the modern world, but are also of fundamental importance to a range of subject areas such as science, engineering, finance, and economics which all increasingly use computer modelling. They also argue that, for reasons of equity and challenging stereotypes, all children should have access to computing education that prepares them for a world of technology that does not yet exist.

Computational thinking teaching for all can also be seen as crucial for widening participation and addressing gender imbalances in the subject. The Royal Society's (2017) critical report on computing education in schools found that computing in the UK was predominantly male, with only 13% of students on full-time computer science degrees being female. The most recent figures from UCAS (the UK University Admissions Service), however, show that this figure is improving, with computing now the 7th most popular subject for UK 18-year-old applicants and a 113% rise in applications from women (BCS The Chartered Institute for IT, 2024). The fact that all children in England should have been taught computing since 2014 could have affected this. Being taught computing at school is also likely to impact wider participation, giving underrepresented or minority groups access to the subject, who may otherwise not have considered it open to them (Guzdial, 2015; The Royal Society, 2017).

Alongside future career benefits, several studies have found that learning CT can positively impact pupils' attitudes to learning and learning behaviours. These studies, however, are often based on teacher reports and need more substantial evidence to support their claims. Bell, Duncan and Atlas (2016), for example, when examining teacher responses to teaching primary computing in New Zealand, found that teachers regularly mentioned increased teamwork, communication and cooperation among young learners taught CT, suggesting that computing was helping to develop children's social skills. This was especially true with children who did not typically contribute and commonly found social skills difficult. Although conducted within the New Zealand education system, primary teachers there face similar challenges to those in England when integrating CT into their curriculum, making the findings informative for understanding how generalist teachers adapt to new computing expectations. Allsop (2019) and Faloon (2016) also mentioned increased cooperation and collaborative skills as additional benefits to their studies, whilst Tran (2019) also mentioned increases in problem-solving. However, these skills are not CT-specific and could have been developed in other areas.

There is some evidence for the benefits of learning CT, but this is not universally accepted. Tedre and Denning (2016, p. 125), in their history of computational thinking, argue against claims that CT should be seen as 'the best method of thinking and problem-solving' but state that it should be viewed alongside other methods of thinking and problem-solving, such as 'engineering thinking, science thinking..., logical thinking, rational thinking... critical thinking' as part of a broader range of valuable skills. Wing (2010)

herself recognised that computational thinking draws on various other thought processes, including mathematical and engineering thinking, and overlaps with logical and systems thinking. If this is the case, we can question the uniqueness and widespread application of CT.

Tedre and Denning (2016) also questioned whether CT has full relevance beyond computer science, as some claim. They argue that similar concepts could be taught through other methods, disputing the claim that problem-solving skills developed through learning CT are transferable to other non-computational areas of learning. Guzdial (2015) also disputes the claim that students can transfer their knowledge from computer science to their daily lives, as students often do not change their problem-solving behaviour to use CT outside of computing lessons.

Not all authors or teachers agree with incorporating CT into primary education. Indeed, there was plenty of opposition to the inclusion of computing in the National Curriculum when it was implemented, with members of industry arguing that, whilst not everyone needs to understand how computers work, they do need to be digitally literate, meaning that the focus of the curriculum should be on improving digital literacy, rather than on abstract computing skills (Larke, 2019). Whilst the original draft for a broader ICT curriculum was written in consultation with industry members, teacher educators from universities and computer science teachers, the final draft was heavily influenced by industry, leading to a sense that 'teachers and their interests had been betrayed.' (Larke, 2019, p. 1141). This perspective can influence how, or even whether, computing is taught at all. This will be

discussed further in Section 2.2.4, which provides a more detailed examination of teachers' perceptions of the computing curriculum and CT.

There are many arguments for the universal teaching of computing; however, whether CT should be taught specifically is more contentious. Some authors argue that it should be taught separately, while others feel the concepts are too abstract for young children. It is, however, a mandatory part of the English National Curriculum (DfE, 2013b). Computational Thinking should be taught in all schools from Key Stage 1; the next section will, therefore, discuss the best age to introduce it to children.

2.2.2 When should Computational Thinking be taught?

Computational thinking is included in the English National Curriculum (DfE, 2013) from Key Stage 1 (ages 5-7), but this is not the case in other countries. This section will examine the arguments regarding the optimal age for introducing concepts and practices related to CT in the classroom.

Earlier research on CT practices in schools primarily focused on high school students; however, research has increasingly explored the introduction of CT to younger children. Ching and Hsu's (2023) literature review of CT in schools worldwide found that half of the studies on educational robotics (ER) from 2012-2021 involved children in Pre-Kindergarten to 2nd grade (ages 3-8). Sullivan and Bers (2016) conducted a classroom-based study using educational robotics with children from Pre-K through second grade (3-8) in the USA. They found that while younger children needed longer to develop critical concepts, older children moved past foundational concepts quickly and

began experimenting with sensors, using repeat loops and conditionals to create more complex programmes for their robots. These studies have implications for the thesis as English primary teachers introduce CT concepts at similar developmental stages, making their findings useful for understanding how younger children engage with robotics and foundational CT concepts. Complementing this, Tran's (2019) research with third-grade pupils (ages 8-9) in the USA highlights the benefits of early exposure to CT for promoting positive attitudes towards computer science, developing problem-solving skills, and encouraging longer-term engagement with the subject. These results are likely to generalise, and so by fostering early confidence and interest, we may help address early participation inequalities, including gender imbalance, as children transition to secondary education in England.

Pérez-Marín, Hijón-Neira, Bacelo, and Pizarro (2020) working with 9-year-old children in Spain using Scratch suggest that it is possible to teach basic computer programming concepts (including memory, programming, conditionals, and loops) and improve their CT with children as young as nine. Although conducted in a different European country, the study suggests that pupils in Upper Key Stage 2 are developmentally ready to engage with core programming ideas when introduced through accessible, block-based environments. They argue that introducing the concepts early leads to greater understanding and enjoyment than among secondary or university students, supporting the rationale for early CT exposure within the English computing curriculum. Supporting this emphasis on early engagement, Falloon's (2016) research in New Zealand introduced even younger children, five- and six-year-

olds, to basic coding through Scratch Jr. on the iPad. His research revealed the emergence of basic CT skills and the development of cooperation, collaboration, and self-management skills in young students. Together, these studies strengthen the case for introducing CT and programming concepts early in primary school, suggesting that young children are capable of meaningful engagement and that early exposure may have both cognitive and broader developmental benefits.

Bell and Duncan (2018) summarise a range of positive consequences for introducing children to computing and CT in primary school. These include exposure before misconceptions about the nature of the subject can develop, increasing the diversity of students studying computer science, empowering children to become creators rather than users of technology, and earlier exposure, which makes it easier for children to learn programming languages. Although they make some valuable points, as a core textbook, we need to know the nature of their evidence to rely on their conclusions entirely.

These studies suggest that, with the right pedagogical approach, it is possible to develop some CT skills in even the youngest primary school children (ages 4-5), with positive benefits for introducing basic CT as early as possible. However, for concepts to be fully understood, they must continue to be built on. The following section will explore how children are taught computational thinking in schools.

2.2.3 How are children taught Computational Thinking in schools?

There are several ways to teach CT in schools, with the most extensive research conducted in computer science and programming. In fact, Kite et al.'s

(2021) worldwide thematic literature review found that 81% of studies (65 out of 80) used programming as the primary method of teaching CT, with only six studies (8%) using 'unplugged' approaches; most of these studies focused on children's problem-solving. Some teaching methods include using virtual or block-based programming languages, such as Scratch or Alice; Educational Robotics, such as LEGO or Bee-Bots; physical computing devices, such as BBC micro:bit; and teaching through unplugged activities. This section will explore these different approaches to teaching CT before discussing where the teaching of CT should be focused in the curriculum.

Educational robotics, such as LEGO Mindstorms, KIBO and Bee-Bots, are growing in popularity for introducing CT concepts to younger children and can be used to create 'collaborative, interactive, and engaging learning environments' (Ching & Hsu, 2023, p. 10). Catlin and Woollard (2014) point out that ERs are not new in schools, as teachers have been using programmable toys for over 30 years. There has, however, been increasing interest in studying their use to teach CT. For example, Angeli and Valanides (2020) used Bee-Bots with five and six-year-olds in Cyprus. They found that, with scaffolding, even young children could decompose a problem and improve their spatial awareness with the appropriate support, suggesting that this also might be possible in England.

Sullivan and Bers (2016, p. 4) also worked with robotics through an 8-week classroom-based project with pre-kindergarten to second-grade children. They suggest that robotics can assist with young children's cognitive, fine motor and social development whilst teaching them about the types of 'sensors and electronics' they encounter in daily life in a hands-on and engaging way.'

However, Chevalier, Giang, Piatti, and Mondada (2020) argue that without sufficient instruction, lessons involving ER can become more of a trial-and-error approach than a means of developing problem-solving strategies. This can limit skill development, meaning CT processes must still be developed. These studies suggest that ER can be valuable for developing CT practices, especially in younger children. However, scaffolding is required to ensure children connect to CT concepts.

Another popular method of introducing CS to children is through the use of a variety of visual (or block) programming languages (VPL), such as Scratch, Blockly or Kodu, with Scratch being the most popular program used (see, for example, Allsop (2019), Kwon, Ottenbreit-Leftwich, Brush, Jeon, and Yan, (2021) and Kwon et al. (2023)). Visual programming languages can reduce barriers to programming by allowing novice programmers to assemble blocks of code, typically via a drag-and-drop interface, without learning complicated text-based languages or worrying about typing errors. These ‘pupil-friendly’ approaches are regularly used in primary schools, with The Royal Society’s (2017) influential report finding that 38% of primary school teachers preferred the programming language Scratch, whilst 17% used Logo or Kodu (15%). Secondary school teachers were much more likely to use a variety of text-based languages, with 21% using Python, 10% using Java, and 6% using Visual Basic as their preferred programming language. However, 19% still preferred Scratch.

Kwon et al.’s (2021) problem-based CS Scratch curriculum for lower-secondary aged students suggests that this approach can lead to increases in self-efficacy, positive attitudes towards CS and confidence in CT, but

particularly where students have prior knowledge of CT (2021). This suggest that foundational CT understanding, typically built during primary schooling, can influence pupils' confidence and progression in later years. In further analysis, Kwon et al. (2023) analysed pupils' Scratch projects, finding that children had developed their understanding of CT concepts such as sequences and events, but that their programmes were very detailed and showed little knowledge of abstraction, one of the most fundamental concepts in CS. This suggests that careful curriculum design and explicit instruction in abstraction may be needed for children to succeed in primary school.

Allsop (2019) examined the games created in Scratch and Alice by Year 6 (age 10-11) primary school pupils from one class in a large London primary school. Her research identified that while nearly all the games written included key algorithmic concepts such as sequences, loops, and conditionals, there was a need for more evidence of abstraction, as children often created complicated sequences. Her findings revealed that whilst all of the children could develop a set of instructions, with the majority able to have two things happening simultaneously (parallelism), only one child could introduce variables into their design. Allsop (2019) also discussed the learning behaviours (or approaches) used by the children, finding evidence for all of the approaches identified by Grover and Pea (2018); in particular, the problem-solving approach used had children collaborating on their game designs and showing perseverance when debugging coding errors or solving problems related to game design. This study, however, involved significant researcher involvement in one classroom; it is difficult to generalise this to English primary schools as a whole to see whether children taught by non-

specialists use CT concepts in the same way. The fact that the games were created in Scratch may also mean that the children focused more on game design than on the CT processes themselves.

Another English study also focused on teaching with Scratch, integrating mathematics and computer science through the Scratch: Maths (SM) project. This large-scale study developed a 2-year intervention with 9-11-year-olds in four London Primary schools to build mathematical knowledge through programming (Benton et al., 2016). The intervention employed a combination of tasks within the Scratch environment and unplugged activities (Clark-Wilson et al., 2019), with activities designed around core computational ideas that were then used to explore mathematical concepts and enhance mathematical reasoning. The researchers found high levels of engagement with the activities (Benton, Hoyles, Kalas, and Noss, 2017). However, while the pupils engaged with Scratch, it was difficult to identify progress in CT, as algorithms were deemed a problematic concept for both children and teachers, and more focus seemed to be on understanding mathematical rather than CT concepts.

These studies, combined, show that while Scratch is very popular and engaging for pupils, careful planning is required to ensure that they make connections between what is happening on the screen and key CT concepts. The block-based nature of systems like Scratch may also make it more challenging for pupils to relate to real-life computing scenarios.

It is possible to combine text-based systems with ER. Berland and Wilensky (2015) conducted an earlier study comparing the use of virtual and

physical media (utilising the VBOT learning environment) to teach computational thinking to middle school students in America. Students taught through a physical medium were more likely to approach a problem from the bottom-up perspective of an individual robot, whilst those working in the virtual environment approached a problem from a group or team perspective. The authors concluded that a curriculum that allows students to engage with virtual simulations and physical manipulatives would yield the greatest gains in students' development of computational thinking. Many of the same pedagogical tools (such as robotics kits and simulation software) are used in English primary schools, and the distinctions identified between virtual and physical engagement offer insight into how pupils experience different models of CT instruction. However, the feasibility of such approaches will vary considerably across English primary settings, as not all schools have access to physical robotics kits, and integrating these resources into existing infrastructure can present practical challenges.

Another growth area in CS education is small programmable devices (or microcomputers) such as Arduino, Raspberry Pi and the BBC micro:bit (Hodges et al., 2020). These physical computing devices can utilise various inputs but often involve visual or text-based programming combined with a physical interactive device that may have sensors and a variety of outputs. Hodges et al. (2020) argue that these can be viewed more positively from a student's perspective. They are more open-ended and can stimulate creativity, enabling students to pursue personally meaningful projects and goals. BBC micro:bits in particular have been used in English secondary schools with children in Key Stage 3 (KS3), as a large-scale project saw around 800,000

devices funded and given to secondary schools in 2016 (Sentance, Waite, Hodges, et al., 2017) and further devices given to code clubs in primary schools.

Sentance et al.'s (2017b) qualitative study explored the responses of English Year 7 (aged 11-12) pupils to the use of micro:bits soon after they were introduced. They found that the novelty of the physical nature of the devices was highly motivating to pupils in the classroom and highlighted four key benefits of pupils working with physical computing devices: **motivation** due to a visible outcome; **tangibility** allowing students to make natural connections with concrete understanding; **collaboration** as devices lead to group work and **creativity** through students relating to the physical nature of the task, unleashing creativity and strengthening engagement.

Kalelioglu and Sentance (2020) also found micro:bits to be positive from a teacher's perspective, as they solve problems associated with physical computing. They are cheap to buy, easy to use without much preparation, and have a block-based editor similar to Scratch, making them accessible to younger children. They also highlighted the importance of instant feedback when pupils run code to help children debug their work. Although much of the research to date has focused on secondary contexts, the affordability and block-based programming environment mean that micro:bits are increasingly being used in Upper Key Stage 2 (UKS2) in England, suggesting strong potential for broader adoption in primary computing. This notable gap warrants further investigation.

The research examined so far requires some form of technology, either through educational robots, programmable devices, or visual or text-based programming. In contrast, Bell and Duncan (2018) argue that the value of working with computational ideas goes beyond programming. As such, it should be taught through areas other than programming, including unplugged activities, cross-curricular approaches, and other subjects.

Bell et al.(2009, p. 20) originally proposed an ‘unplugged’ approach to teaching CT, using non-computer-based activities and games to explore the thinking computer scientists use. Unplugged activities, which teach computing concepts away from the computer, can be seen as a way to educate pupils without requiring a wide range of resources and removing the distractions that can occur in a computer suite (Curzon et al., 2014). Originally intended as an outreach tool, an unplugged approach aims to engage students through a meaningful, age-appropriate approach (Bell & Lodi, 2019). An unplugged approach on its own, however, may not be enough; Bell and Lodi (2019, p. 350) go on to state that to be most effective, unplugged approaches need to be used thoughtfully and that they are most effective when they are used in combination with a 'plugged in' approach, allowing students to make connections from the unplugged learning to a using a digital device. While the approach has been popular and can enable teachers with limited resources to cover the curriculum, its actual impact on the understanding of CT has been disputed (Guzdial, 2015). Taub et al.(2012) explored an unplugged approach with two classes of middle-school students (age 12-13) in Israel and found that although pupils showed a modest improvement in their understanding of CT concepts, the approach was associated with a decrease in students’ interest in

pursuing computing further. The students' age may be a factor here, as they may struggle to make the connection between the unplugged approach and a career in computing. While the study sits outside the English primary phase and was conducted in a different national context, it nevertheless offers a useful insight, suggesting that unplugged approaches alone may not sufficiently sustain engagement or convey the relevance of CT. Taken together with wider evidence, these findings suggest that unplugged methods can be valuable for introducing particular CT ideas, but are most effective when combined with hands-on, practical computing experiences that help learners make clear connections between the activity and the underlying computational concepts.

The reviewed research has addressed methods for teaching CT to date and will now consider where in the curriculum it is most appropriate to introduce it. Several authors argue for a cross-curricular approach to teaching CT: Bell, Duncan and Atlas (2016), for example, argue for an 'integrated' curriculum, suggesting integrating CT into other subjects can enhance learning across a range of subjects, making learning more suited to a digital world. Bell and Lodi (2019) also argue for a cross-curricular approach, suggesting that CT and CS should not be seen as ends in themselves but rather applied in practical contexts. This cross-curricular approach may help teachers integrate computing into an already crowded curriculum, enabling students to see its relevance in their everyday lives.

Much of this research has been conducted in the Science, Technology, Engineering, and Mathematics (STEM) disciplines, which, as Weintrop et al. (2016) note, increasingly involves computation. They argue that integrating

CT into maths and sciences classrooms will better prepare students for careers in these areas, giving them a more realistic view. They also suggest that this approach can help CT reach a wider audience and encourage students to study CT for longer. Luo et al. (2020) found that combining computational thinking concepts with robotics to teach life cycles to US elementary school girls (ages 8-10) enhanced the understanding and enjoyment of both. However, as this study involved only two participants, it is challenging to draw firm conclusions.

Several researchers argue in favour of teaching CT through mathematics. For example, Sanford and Naidu (2016) suggest using Excel, alongside mathematical word problems, to allow students to go beyond just solving a problem and ask 'what if' questions. As previously discussed, the Scratch: Maths project (Benton et al., 2016, 2017; Clark-Wilson et al., 2019) advocated for combining the teaching of mathematics concepts with computational thinking, although the emphasis here seems to have been more on the mathematics.

Wolz et al.(2011) taught computational thinking through interactive journalism in contrast to the previously mentioned studies. They specifically targeted students who considered themselves 'not math types' and underrepresented groups. Their study involved a week of training teachers, who then worked with middle school students from their schools on creating videos and procedural animations using Scratch to illustrate news stories. The study found that engaged language arts teachers were eager to incorporate the interactive journalism model into their classroom instruction; this, in turn, positively impacted students' perceptions of programming as enjoyable and

accessible to anyone. The teachers, however, were not setting learning goals related to computational thinking; the animations and videos produced may have needed to reflect computational thinking skills, making it difficult to assess whether the students were becoming more proficient in computational thinking.

All the studies show that children enjoyed learning CT, whether taught through computer science or other subjects. However, it is essential to consider who is teaching CT in schools, as researchers often presented their own teaching in many of these studies. The following section provides a more detailed examination of this topic.

2.2.4 What are teachers' perceptions of teaching computing and Computational Thinking?

Notably, while plenty of studies research how children learn computational thinking, relatively few examine this from the classroom teachers' point of view (Rich et al., 2019).

A few studies have examined teachers' views of the English computing curriculum. Sentance and Csizmadia (2017) surveyed over 300 teachers in 2014 as the curriculum was being introduced. Whilst their self-selecting sample was generally confident about teaching computing, they reported a range of challenges to the introduction of the new curriculum, including teachers' lack of subject knowledge, students' lack of understanding of the content, differentiation to meet different levels of ability, and students' willingness or ability to problem-solve. The fact that the teachers who responded to their survey were generally engaged in computing but reported

various challenges suggests that these challenges might be even greater among less confident teachers. An extensive BT and Ipsos MORI (2016) survey found that while 97% of teachers felt it was their job to prepare children for the digital world, only 25% strongly agreed that they could. When specifically asked about teaching CT, 57% reported understanding what CT was, but 35% still required help teaching it, and 31% did not teach it in computing lessons or the broader curriculum. These worrying statistics indicate that many teachers lack confidence in teaching the computing curriculum effectively in their classrooms, with some opting not to teach it due to its perceived complexity.

The Royal Society's (2017) extensive survey, conducted after the introduction of the new curriculum, found that many teachers in England believed that the Government changed the subject (of computing) without providing them with the necessary support to teach it effectively. A large percentage of respondents indicated they were more confident in teaching the original ICT parts of the curriculum. Sentance and Waite (2018) also noted a need for pedagogical guidance as a concern for teachers; they discuss a 'pedagogical vacuum' filled by a 'plethora of approaches' that were not necessarily supported by research.

More worryingly, Larke (2019) reports that whilst confident teachers will teach content that meets their professional standards, teachers in her study did not believe that the English National Computing Curriculum (2013) did this and largely neglected it in favour of other subject content. The teachers viewed the content of the programme of study as follows:

“Developmentally inappropriate, too narrow, too difficult to teach without additional staff or technological resources, requiring more prerequisite knowledge than their students had or could be expected to acquire during the school year, or as a lower priority subject compared to other subjects” (p.1138).

This damning view meant that, three years after the introduction of the curriculum, computing should have been taught weekly, but it was not. Computing was rarely taught, with teachers choosing not to teach a subject they felt they needed more resources or external motivation to teach. The UK Government’s 2023 survey of teachers’ working lives reported that only 39% of primary school teachers felt confident about teaching computing (DfE, 2024c) showed that this was still the case nearly 10 years after the curriculum was introduced. This lack of confidence is significant, as it will likely affect how, or even whether, computing is taught and needs to be addressed for it to be taught successfully.

September 2024 marked 10 years since the introduction of the 2014 curriculum; however, the picture in secondary education remains largely negative. The SCARI Computing Project, a large-scale project on the shift in secondary computing in England (Kemp et al., 2024) found that at the secondary level, the transition from ICT to Computing presented challenges, including a dramatic decline in the number of girls studying computing at GCSE level (only 21% in 2023, compared to 43% studying ICT in 2015). While the overall numbers of students studying computer science at GCSE and A-level have risen, there has been a ‘general decline in computing and digital skills education’ (pg. 12), especially among girls, ethnic groups and

students from underserved socioeconomic backgrounds. Computing also faces ongoing challenges in recruiting secondary school computing teachers, as well as a decline in the overall number of hours spent teaching computing in schools (Kemp et al., 2024). The following section will explore additional factors that can influence whether and how computing, and therefore CT, is taught in schools.

2.3 Factors that prevent the teaching of computing and the use of educational technology

Alongside teachers' perceptions of teaching computing, additional challenges to teaching computational thinking in schools must be considered. Whilst few studies directly examine why computing is not being taught, studies into technology use can give insight into factors that may affect the teaching of computing. If teachers are reluctant to integrate computers (or educational technology) into their teaching, they are unlikely to teach computing. This next section will examine why computing is not being taught and why technology may not be used effectively in schools.

Ertmer's (2005) seminal research distinguishes between first-order and second-order factors that influence teachers' integration of ICT into their teaching. First-order barriers are considered extrinsic to the teacher and include 'lack of resources, time, access and technical support' (Prestridge, 2012, p.449). First-order barriers are considered easily removable once funding is provided and are therefore usually the first barriers to focus on during the reform of ICT use (Donnelly et al., 2011). Second-order factors are intrinsic to the teacher, rooted in teachers' core beliefs, and revolve around

'teacher-student roles, teaching methods, organisational (sic) and management styles and assessment types' (Donnelly et al., 2011, p.1470). Second-order factors are much more likely to impact future change, so they are the most significant and the hardest to alter.

2.3.1 First-order Factors Affecting Technology Integration

Although Agyei and Voogt (2011) state that computer access is no longer a relevant issue in many developed countries, the introduction of the computing curriculum in 2014 led to new hardware and software resource requirements. In fact, the original 2012 Royal Society Report that preceded the curriculum changes cited school infrastructure as one of the most significant barriers to the effective teaching of computing (Furber, 2012). Many UK Primary schools replaced their ICT suites with banks of laptops or iPads that often need to be booked out or timetabled, leading to resourcing issues. However, this technology investment needs to be updated after ten years. Sentance and Csizmadia's (2017) large-scale survey of teachers in England, published three years after the introduction of the new curriculum, found that both primary and secondary school teachers reported challenges related to not having sufficient hardware or software to teach computing, as well as problems with out-of-date networks in schools that made it difficult to run programs. They also discussed the need for more funding to purchase resources for teaching what was then a new subject. This does not seem to have improved in the following decade, with 95% of primary leaders questioned as part of the UK Government's 2022-2023 Technology in School Survey, citing budgetary constraints as a barrier to increased technology adoption in schools (IFF Research, 2023).

The availability of computing resources significantly impacted technology use in schools. Wong (2016), in a study investigating the uptake of educational technology among primary school teachers in Hong Kong found that facilitating conditions (such as resource availability, school funding, technical support, and time) were the dominant factors influencing whether teachers used technology. Although conducted in a different national context, these findings resonate strongly with the English primary setting, where variation in hardware, infrastructure, and support staff similarly affects teachers' confidence and capacity to embed computing and digital tools in classroom practice.

Teachers working in schools in England may have been considered to have high access to educational technology. The TIMSS data from 2011 (Sturman et al., 2012) stated that 90% of English children in year 5 had access to at least one computer for every two children, the highest level reported by any of the participating TIMSS countries. However, by 2019, this figure had fallen dramatically, with only 24% of students having access to computers they could sometimes use (Mullis et al., 2020), below the international average of 29%. Even more worryingly, the UK Government's 2022-23 survey of Technology Usage in schools found that the proportion of schools reporting no laptops or tablets available for pupil use had increased from 2021 (IFF Research, 2023). The study further reported that whilst 91% of schools reported having laptops available for pupil use, 44% reported that they only had them for less than 25% of pupils. A similar picture was observed for tablet availability: 50% of schools reported having tablets available for fewer than 25% of pupils, with a ratio of 1 tablet per 7 pupils (IFF Research, 2023). If, as

the report suggests, 47% of primary pupils are using 'end-user devices' such as 'desktop computers, laptops, tablets, smartphones or other mobile devices' in less than 10% of lessons, with 80% of pupils either not using them at all or using them in less than 25% of lessons, then it is difficult to see how computing can be taught at all (IFF Research, 2023, p. 48).

The rapidly changing nature of educational technology could also impact the relevance of school resources. Petko (2012) agrees with this view, stating that when investigating technology use in schools, it is necessary to consider not only the average number of devices available for teachers to use with students but also 'whether these devices ... (sic) are up-to-date and well maintained, and the ease and flexibility of computer access at the school.' (p.1352). With budgetary constraints and the high costs of new technology, meaning that it is unlikely to be replaced, it is crucial to work with what is currently available. This is particularly significant when implementing change in schools, as there is no point in planning activities that require a substantial outlay of money or utilise specialist technology if this will not be available.

It is also necessary to consider technical support in schools, as devices can be challenging to integrate into often unreliable school networks. The 2022-23 UK technology in schools survey (IFF Research, 2023) found that only 41% of schools had a reliable broadband connection, only 29% reported reliable Wi-Fi throughout the school's premises, and only 36% reported that the school's network provided a fast and reliable connection.

Alongside technological problems and resource limitations, time must also be considered when addressing the challenges of teaching computing in

schools. Clark-Wilson et al. (2019) recognise the challenge of allocating curriculum time to CT, with three-quarters of the English schools they worked with struggling to cover the planned work; this was mainly a problem in Year 6 (ages 10-11), where high-stakes National Curriculum tests meant that a large proportion of class time was spent on revision and ‘teaching to the test’. This suggests that computing teaching needs to be easy to integrate into the school day.

Clark-Wilson et al. (2019) also mention internal staff changes and teacher movement in and out of school as concerns, as teachers who had engaged in their Continuing Professional Development (CPD) were often not teaching the CT sessions. Teachers need to have the opportunity to engage in ongoing professional development throughout their teaching careers, particularly in subjects such as computing that change rapidly. The UK Government’s 2022-23 survey of educational technology uptake in schools, however, found that 67% of school leaders cited not having enough time for CPD, with 65% citing cost, as being to some or a great extent a barrier to increased uptake in schools (IFF Research, 2023). These considerations need to be taken into account when developing CPD. Section 2.4 will discuss these issues further and examine the development of CPD with teachers.

These first-order factors, therefore, are likely to have a significant impact on the teaching of computing and on the use of educational technology. Access to computers, new resources, and teaching time are all equally important for developing computational thinking. If teachers struggle to book time in computer suites and have to use outdated resources, they are significantly less likely to devote time to teaching computing.

2.3.2 Second-order Factors Affecting Technology Integration

Teachers' beliefs about ICT as supportive of (and transforming) teaching and learning can impact technology integration in the classroom, with positive beliefs encouraging higher classroom integration and negative attitudes discouraging it. These second-order barriers to change are further affected by teachers' views about classroom management and teacher-student relationships (Donnelly et al., 2011).

Ertmer (2005, p. 27) states that 'ultimately, the decision regarding whether and how to use technology for instruction rests on the shoulders of classroom teachers.' Therefore, if educators are to achieve meaningful, or second-order, changes in classroom teaching practices, they must examine the teachers themselves and their beliefs about teaching, learning, and technology. As teachers' beliefs influence their practice and teaching behaviours in the classroom, a change in belief regarding the usefulness of ICT in the classroom is seen as permanent, requiring a shift in established pedagogical beliefs. This can either happen as a 'collision', adding onto existing established curriculum practice or as a 'collusion' affecting change to practice (Prestridge, 2012, p. 449).

In reviewing the research on the relationship between pedagogical beliefs and technology use, Tondeur et al. (2017) found that teachers with more teacher-centred beliefs were less likely to see technology as being important in the teaching and learning process. However, as positive experiences with technology could have an impact on these pedagogical beliefs, they concluded that regardless of teachers' pedagogical approaches, technology should be introduced in ways that align with current practice and,

therefore, 'appeal to their values and increasing (sic) the likelihood that teachers will integrate and use technology' (p. 569). It is thus essential that teachers see the use of technology as beneficial to both themselves and their students.

Peer and management support can be essential to help teachers overcome second-order barriers to technology integration. Durff and Carter (2019) found that elementary school teachers in the USA overcame attitudinal barriers to technology integration through training and peer discussions. Formal and informal peer mentoring, along with shared planning, enabled teachers to overcome negative attitudes toward computer use, ultimately leading to the successful integration of technology. Although conducted in an American context, these findings are relevant to English primary schools, where similar patterns of support, particularly peer collaboration, shared planning, and encouragement of leadership, are known to influence teachers' confidence and willingness to embed computing and educational technology into their practice. Ottenbreit-Leftwich et al. (2018) also found that strong mentorship positively impacted technology use and teachers' ability to overcome technology-related challenges. If the long-term goal is to increase the quantity and quality of computing teaching, then support from the school management team, especially the school's computing coordinator, will be essential.

Teachers with negative attitudes towards educational technology and teacher-centred pedagogical beliefs may share similar attitudes and feelings towards teaching computing or CT. This would mean that teachers who do not utilise educational technology are less likely to incorporate computational

thinking into their teaching. The following section will further examine how teachers' self-efficacy regarding the use of technology and the teaching of computing can affect how, or whether, the curriculum is taught.

2.3.3 The impact of teachers' self-efficacy on computing teaching

Self-efficacy, defined as an individual's belief in their ability to accomplish a goal, can impact teaching in the classroom. Bandura's (1977) model of self-efficacy explores how beliefs about ability impact how well people perform in a particular situation and can be used in educational research and training when considering the development of a new or higher level of skill (Mannila et al., 2018).

Teachers' self-efficacy can be seen as a teacher's belief in their ability to successfully cope with the tasks, obligations and challenges related to teaching, for example, managing behaviour and giving instruction (Barni et al., 2019). Teachers with a high level of self-efficacy are more likely to view challenges as opportunities for growth and to engage with them. However, teachers with low self-efficacy may view challenges as signs that they cannot do something and, as such, may give up early to avoid a perceived threat to their competence (Zhou et al., 2020). Whilst Bandura's (1977) original model was a generalised concept, later models have become more context-specific, such as computing or ICT self-efficacy. Computing self-efficacy refers to an individual's confidence in their ability to use computers to manage tasks or solve problems effectively. Many studies (such as Kwon et al., 2019; Zhou et al., 2020) have explored the impact of computing self-efficacy on teachers' technology usage. Whilst this is not directly related to the current study, the parallel literature has implications, as teachers with higher computing self-

efficacy, who already use technology in the classroom, may also be more likely to have higher self-efficacy for teaching computing and be more open to teaching computational thinking.

Kwon et al. (2019) examined the impact of self-efficacy on whether middle school teachers in the USA integrate mobile computing devices, such as iPads, into their teaching. Their survey concluded that teachers with higher technical skills had higher computing self-efficacy and were therefore more likely to integrate mobile technology into their teaching practice. Male teachers were found to have higher self-efficacy in using technology than female teachers. However, gender was not found to affect teachers' beliefs about the usefulness of mobile devices in teaching. The survey's relatively low response rate (57 teachers, or 5% of those approached) and the fact that respondents self-selected mean that only teachers who use iPads may have responded, which could affect the results. Although this study is situated within the US context and involves older learners than those in English primary schools, it remains relevant because similar patterns of variability in teacher self-efficacy can be seen in England, where confidence with digital tools stringly influence teachers' decisions to adopt computing-related technologies in the classroom.

A further important finding from Kwon et al.'s (2019) study was that years of teaching had a significantly negative relationship with teachers' technical skills, thus implying that more experienced teachers were less likely to use technology when teaching. This could be linked to age, with younger teachers having higher technical skills. As technology changes rapidly, it is expected that the technology that more experienced teachers were introduced

to during training would need to be updated; this argues for continuing CPD to ensure that teachers remain up-to-date and more likely to integrate technology into their teaching.

Mannila et al. (2018) explored the links between primary school teachers' self-efficacy and digital competence during the introduction of computing in the Swedish National Curriculum. They found that teachers reported the highest computer self-efficacy in information and data literacy skills, such as searching for information and communicating online, areas that had been part of computing practice prior to the reform. In contrast, teachers expressed significantly lower confidence in the newer elements of the curriculum, including digital content creation and writing programs for problem-solving. This pattern is highly relevant to the English context, where the transition from the former ICT curriculum to the 2014 computing curriculum also required teachers to move from familiar digital literacy practices to less familiar areas such as algorithms, programming and computational thinking, meaning that the same confidence gap is likely to exist. Mannila et al. (2018) further found that teachers with the highest self-efficacy across all areas were most adaptable to new situations, whilst those with low self-efficacy were low across all regions. This distinction is significant for computing in English primary schools, as teachers with stronger general self-efficacy may be more willing to engage with teaching the newer, more conceptually demanding parts of the curriculum.

While previous studies have explored self-efficacy related to technology use or digital competence, Sen (2023) specifically examined the relationship between pre-service teachers' self-efficacy for learning and CT

skills. Their large-scale (482 pre-service teachers) quantitative study found that Turkish pre-service teachers' CT skills were directly affected by their self-efficacy for learning and performance. The authors argue that pre-service teachers with a higher self-efficacy for learning were less doubtful about themselves, had more positive expectations as problem solvers and, therefore, may be more willing to engage in learning CT. Although conducted in Turkey and focused on pre-service teachers, the study remains relevant to the English context; many English primary teachers also enter the profession with limited prior exposure to CT, suggesting that their initial confidence may shape their engagement with the subject. While Sen's (2023) study did not directly examine future teaching behaviour, the findings suggest that developing strong learning self-efficacy during ITE could play an important role in supporting teachers' confidence to teach CT in the classroom.

Building on the idea that self-efficacy is shaped not only by individual beliefs but also by the professional environment in which teachers work, wider research has examined how school-level factors influence teachers' confidence and engagement. School leadership can significantly impact self-efficacy. Gümüş et al. (2022) conducted a large-scale quantitative survey of 427 teachers without formal leadership roles in Turkey. They found that the stronger the teacher leadership in the school, the more likely teachers were to engage in workplace professional learning, share practices, and report higher self-efficacy in their instruction, classroom management, and student engagement. Although the study context in Turkey makes this less applicable to England, the relatively informal nature of leadership is similar to that seen

among subject leaders in English schools, suggesting that strong subject leaders can positively impact how a subject is taught in a school.

Given that teachers' experiences and curriculum expectations differ across countries, it is particularly valuable to consider UK-based research (when available) alongside international studies. Sentance et al.(2017), in their study examining the introduction of the BBC micro:bits to British secondary schools, found a correlation between teachers' confidence and how well they inspired children with physical computing. The study found that teachers who felt most confident in their teaching of physical computing were more likely to embrace change, adapt their teaching to meet their students' needs, and remain undaunted by technology problems. They identified three approaches to teaching with physical computing that could be applied to teaching computing in general: Inspirers, Providers and Consumers.

1. 'Inspirers: Teachers using a range of open-ended activities and taking risks; relating this to pedagogical beliefs
2. Providers: Teachers using structured approaches to learning programming using the device, relating to pedagogical beliefs
3. Consumers: Teachers making use of short, ready-made exercises to motivate and engage; no explicit reference to programming pedagogy' (p. 8).

While consumers will only use short, ready-made exercises in their teaching, inspirers are willing to take risks by giving children open-ended activities. These teachers were also most likely to share resources with other teachers, link computing to real-world contexts and encourage independent

learning. Providers were more likely to use structured resources and focus on learning outcomes in their teaching. These definitions may be helpful when discussing how teachers approach computing teaching in general and when identifying approaches to support teachers through CPD.

If self-efficacy and confidence are impacting computing teaching, then CPD must explore not just how teachers develop their CT skills but also their attitudes and mindsets when teaching computing and using technology in the classroom. The following section will therefore examine how computing has been taught to teachers, both at the pre-service/teacher training level and in schools, to explore best practices.

2.4 Teaching computational thinking to teachers

If the goal is to support the teaching of CT in schools, then teachers will need guidance and education in pedagogies related to teaching CT to Primary Education students in schools (Pérez-Marín et al., 2020). Without this education, Allsop (2019) argues that whilst teachers may be able to describe terminology and teach from lesson plans, they are unlikely to recognise CT skills and concepts in children's work. Pedagogy needs to be CT specific; Angeli and Valanides (2020, p. 3), for example, argue that while educational robots like Bee-Bot are easy for children to use, teachers must learn how to use them 'in an appropriate pedagogical way' to maximise the development of young children's CT skills.

To develop best practices, it is essential to examine previous models of teaching computing to teachers. This next section will therefore explore these aspects, both at the pre-service or teacher education level and among in-service teachers working in schools, to discover best practices.

Much of the earlier research concerning improving teachers' CT focused on enhancing the subject knowledge of teachers already teaching computer science, usually within a high school setting. For example, Blum and Cortina (2007) conducted outreach to develop computational thinking with high school computer science teachers, finding that before their workshop, teachers viewed computer science primarily as programming. Their workshop aimed to raise awareness and develop practical, unplugged activities for teachers to use with their pupils in school, demonstrating that computer science is more than just programming. Morreale and Joiner (2011) also

conducted workshops for in-service and trainee teachers to help them address the step-up in knowledge required from high school to university in computational thinking. Their study reported little development of computational thinking while students were taught C++ and Java alongside Microsoft Office. These workshops also aimed to develop valuable resources for teachers within schools. Practical activities, taught through workshops, were the focus of both studies. However, whether these led to changes in teachers' thinking or long-term changes to what is taught in schools is still being determined.

In contrast to previous studies that focused on developing subject knowledge through practical activities, Zhou et al. (2020) adopted a growth mindset approach when designing a certificate in teaching computing for in-service high school teachers in the USA. A growth mindset can be defined as the belief that your essential qualities can be cultivated through your efforts, strategies and help from others (Dweck, 2017). Zhou et al. (2020) argue that effective curriculum changes depend on teachers' self-efficacy, making it a priority when designing CPD. Their CPD therefore advocated a growth mindset approach from the beginning, with teachers encouraged to focus on self-improvement and to use formative assessment to generate feedback for personal growth. They conclude that high-quality CPD should incorporate active learning, collective participation, and a focus on content, and should be conducted over a sufficient period of time. Whilst this is a valuable approach, their study allocated significantly more time to professional development than is available for the current study in an English context. The fact that the CPD led to a professional qualification means that participants were likely to have

higher motivation and capacity for extended engagement than the generalist primary teachers involved in this research.

While it is helpful to examine how studies with high-school teachers have been carried out, it is important to recognise that the English primary context is fundamentally different. Most primary school teachers in England teach more than 10 subjects (DfE, 2013b) and are rarely computing specialists. As a result, they typically do not enter the profession with the background or subject-specific confidence that many secondary computing teachers possess. Much of the existing research with high-school teachers has focused on developing computing subject knowledge or improving self-efficacy related to programming, rather than on computational thinking specifically. When working with non-specialists who face significant time pressures and competing curricular demands a different approach is therefore required.

Yadav et al.(2014) focused explicitly on developing computational thinking in elementary and secondary teacher education in the US amongst non-specialist trainee teachers. They argue that introducing CT early in ‘teacher preparation’ might allow trainees to ‘see the relevance of CT in their own disciplines’ (p. 54). Computational thinking was taught as part of a compulsory educational psychology course to more than 150 sophomores and juniors enrolled in a teacher education degree program at the university. Computational thinking concepts were introduced through concrete examples from everyday life, relating key terminology to the pre-service teachers' personal experiences, before exploring how CT could be taught in core areas by integrating CT concepts into everyday instruction. 58% of pre-service teachers recognised that teaching CT in the classroom involved teaching

students to problem-solve, and the majority of students (94%) recognised that it could be taught across a range of subjects. However, whilst teaching CT to trainee teachers is imperative, this approach does not address the CPD needs of in-service teachers, or that even recently trained teachers are hesitant to teach CT in the classroom and therefore have ongoing CPD needs.

Bean et al.(2015) also discussed developing CPD for pre-service teachers in the US, focusing on developing a mastery approach to using Scratch. Recognising the pressure of a test culture on classroom practice, they employed a cross-curricular approach, integrating CT into subjects such as music and mathematics. This cross-curricular approach may be helpful in the current study to alleviate time constraints in the crowded school timetable. While their first intervention, using laptop computers, increased students' self-efficacy, they had technical difficulties due to conflicting operating systems when they switched to iPads for their second intervention. This may also be an issue in many English primary schools, which, as previously discussed, have invested in outdated iPads.

Nugent et al. (2022) worked with K-8 in-service teachers in the USA (ages 5-12). They argue there is no point in including text-based programming languages in CPD at this level, as these languages are mainly taught in high school. Their research, which presented CT-focused summer CPD to 53 elementary teachers, found that teachers responded most positively to pedagogical instruction rather than content delivery. However, whilst their CPD increased teachers' knowledge of CT concepts and recognition of the value of the subject to their students, it had no impact on their interest in the subject. This is worth considering because CT may again be relegated to a

time-pressurised timetable, with teachers uninterested in the topic. It would also be easier to engage English teachers in summer CPD with pay; for this study, CPD needs to be taught during the school day or during evening staff development time.

Israel et al. (2015) worked with all of the teachers (20) in one American elementary school to integrate CT into the school's instruction. Starting with a week-long summer school, with follow-up workshops during the winter break, they argue for ongoing professional development and embedded coaching by a technology specialist to increase participation in computing education. Their study found that whilst teachers were generally willing to learn about computing and integrate it into their teaching, even those excited about computing education were nervous about doing so. While their approach to working with the whole school staff is helpful in the present study, it would again be difficult to deliver this level of CPD during the holiday period, as teachers are not expected to engage in formal school activities during this period.

Taken together, these US-based studies provide valuable insights into how both pre-service and in-service teachers can be supported to develop their understanding of CT and computing pedagogy. However, their applicability to the English primary context is necessarily limited by differences in curriculum structure, professional expectations, and the nature of teacher education. In the USA, CPD often takes place during extended summer programmes or as part of credit-bearing university courses, conditions that differ markedly from those in England, where primary teachers work within a more time-pressured system and rarely access such intensive provision. In addition, many US

studies focus on older learners or specialist pathways, whereas English primary teachers are generalists who must teach computing alongside multiple other subjects, often with limited subject knowledge and confidence. These contextual differences highlight the importance of UK-based research to understand the specific challenges faced by English primary teachers and to explore models of support that are both feasible and meaningful within this system.

In contrast to previous studies, Curzon et al. (2014, p. 90) explicitly introduce Computational Thinking to English teachers through unplugged-style computing activities. As previously discussed (Section 2.2.4), unplugged activities can overcome the limitations caused by a lack of space or resources. They worked with teachers after the curriculum was introduced in 2013. They suggested that unplugged activities could be used as an 'explicit teacher training tool' to teach computing concepts to teachers and as a powerful means of introducing CT concepts to adult teachers. Curzon et al. (2014) describe four 90-minute workshops, each aiming to introduce a CT skill/ concept by demonstrating a short whole-class activity that can be used in schools. While most attendees found this approach beneficial, the workshops were very labour-intensive. Therefore, they would be difficult to replicate by one person teaching CPD to a small group. The training was also conducted with a group of self-selected teachers, early in the implementation of the new curriculum, who might therefore be considered to have an interest in computing teaching and a greater level of confidence in their ability to teach computing than primary school teachers as a whole.

In a more recent study, carried out 8 years after the introduction of the curriculum, Leonard et al.(2023) worked with 13 British Primary School teachers to co-create culturally adapted computer science resources for their classrooms. Their one-day CPD workshop was found to significantly increase participants' confidence in adapting resources, suggesting that CPD can be a valuable approach to improve the teaching of computing within a school.

Therefore, as Angeli and Giannakos (2020, p. 3) argue for CT education in schools to progress further:

“Teachers need to be systematically prepared in terms of how to design CT learning activities, how to teach CT, how to assess CT, and how to use technologies to teach CT concepts.”

Taken together, the preceding research highlights certain features that lead to effective professional development for teachers. Teachers need more than just subject knowledge to teach CT effectively; they also require an awareness of how to teach and the self-efficacy to do so confidently. While teachers valued a workshop-based approach and ready-made resources in their classrooms, this approach often did not change teachers' thinking or how CT was perceived.

To work within the current study context, CPD must be designed for in-school and after-school inset sessions. This aligns with Durff and Carter's (2019) research on technology integration, which found that teachers particularly valued training in their classrooms, rather than large conferences, because it enabled them to transfer their learning to working with their students most effectively. The context of the school is important; CPD needs

to be targeted to the Watermill School and its teachers. It is also essential that curriculum change becomes embedded in the school, so CPD must be replicable after the study has concluded, especially if the teaching staff changes. This involves collaborating with school staff to design CPD that is relevant to their needs, enabling them to continue developing. The following section will explore different models of CPD used in primary education.

2.5 Models of School-based CPD

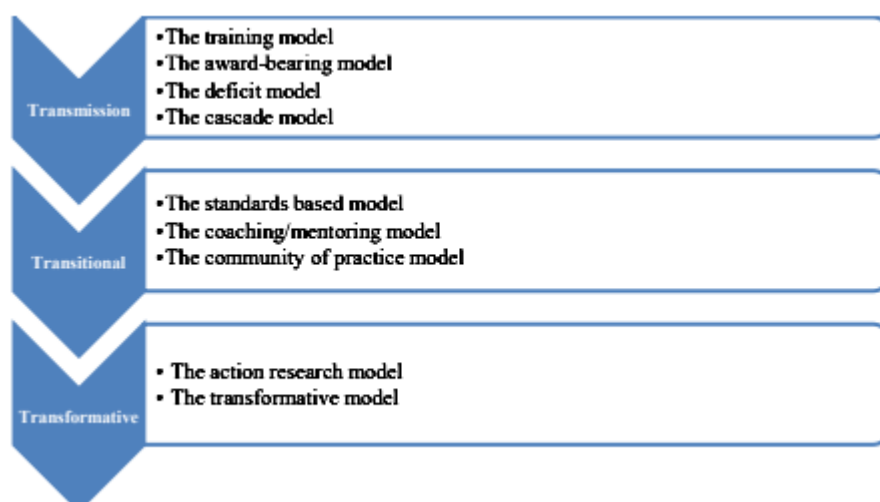
The next section of the literature review will consider models of continuing professional development (CPD) applicable to primary education, followed by an analysis of how these models can be adapted to suit the computing context of the current study.

Kennedy's (2005) seminal framework identified nine key CPD models that might be adopted within schools. Her categorisation and organisation of the CPD models (see Figure 2.1 below, taken from Williams et al. (2022)) argued for increased teacher autonomy, moving from transmission through transitional to transformative approaches. Transmission approaches to CPD are usually 'delivered' by an expert, with participants acting as passive receivers of knowledge. These may be award-bearing (for external validation) or in response to a perceived deficit in a teacher's knowledge. These traditional top-down models of CPD, where teachers passively receive information to implement, are often unsuccessful (McNeill et al., 2016). As

Kennedy (2014) argues, for teachers to make real progress, they need autonomy and the ability and space to enact change.

Figure 2.1

Kennedy's Spectrum of CPD (Williams et al., 2022, p. 29)



Cascade models involve one staff member attending training to pass on knowledge to the rest of the staff (Kennedy, 2005). While these training models are popular, they may not lead to much classroom change because they rarely occur in the classroom, and teachers often lack the time to implement changes. Cascade models are particularly popular in situations where resources are scarce. For example, a subject coordinator may attend a specific training day for their subject and then give a presentation to the rest of the staff (Williams et al., 2022). As this model does not lead to participation, collaboration, or ownership, it is less likely to result in ownership of change, as teachers are less invested in training that they have not personally attended. Key ideas and overall meaning may also be misinterpreted in the process of passing it on.

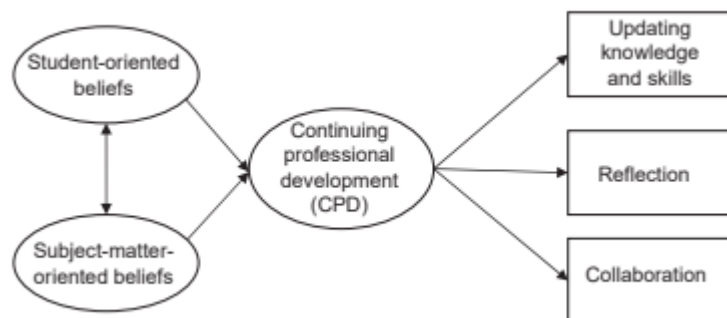
In contrast, transitional models, which Kennedy (2014) later terms ‘malleable models’, include coaching, mentoring, and community of practice approaches. Coaching, a more formal, usually shorter-term approach, where the coach supervises CPD, can be contrasted with a longer-term mentoring, less formal approach, which is usually based on discussion or briefings within an organisation (The Open University, 2019) while community of practice approaches may involve staff working together within an organisation to enact change. These approaches may be suitable for the current study, given the intention to co-develop CPD with a change agent in the school. An external expert could take on either a coaching or a mentoring role. However, Kennedy (2014) also discusses how, whilst there may be increasing professional autonomy throughout the model, this is only transformative if it leads to a positive change in practice. It is also important to recognise that different approaches can yield different outcomes depending on their implementation. Clarke and Pike (2023) examined the impact of forming a community of practice on changing teachers’ attitudes and approaches to primary geography in Ireland. They found that school-based CPD, involving collaboration with all teachers in the school, increased teachers' subject knowledge. Teachers formed a community of practice and identified colleagues to whom they could seek guidance and support. They argue that CPD is most effective when it is collaborative, allowing teachers to share knowledge, ideas, and experiences in a respectful environment.

Teacher beliefs can significantly impact the effectiveness of CPD in changing teachers’ practice. de Vries, van de Grift, and Jansen (2013) examined the beliefs of 260 pre- and in-service secondary school teachers

from the Netherlands, finding a significant relationship between teachers with student-oriented beliefs and CPD. More student-oriented teachers reflected more on CPD, participated more in activities, collaborated more with colleagues, and were more likely to change their practice in response to CPD. While this study is over 10 years old, its theoretical model, illustrated in Figure 2.2 below, can be usefully applied when considering the success of CPD in schools, suggesting that awareness of teachers' beliefs needs to be built into CPD.

Figure 2.2

Theoretical Model of the links between teachers' beliefs about learning and teaching and their participation in CPD (de Vries et al., 2013).



Kennedy (2014, p. 693) later renames transformative models as ‘collaborative professional inquiry,’ which she uses to encompass CPD, including collaborative problem identification and follow-up activities. This will include examining one's practice and may involve engaging with existing research for professional learning. McNeill et al. (2016) investigated how a collaborative model of professional development involving three lead teachers in English primary schools, alongside a university researcher, could enhance

whole-school mathematics teaching. They found that this collaborative approach, in which teachers worked together to design personalised CPD for their schools, was welcomed by teachers and head teachers and showed early indications of student progress. Their collaborative model incorporated elements of various interventions, including classroom observation, consulting with professionals from outside the school, professional dialogue, teacher ownership of CPD, and peer support, rather than a top-down management approach. They further argue for sustained management support for CPD to allow new practices to become established.

Whilst neither of the studies discussed explored the professional development of computing, both employed some form of collaborative approach within a primary school that may be applied to computing and, therefore, to the current study. Using a collaborative approach should lead to the most sustainable long-term change. As a professional from outside of the school, acting as a consultant to help develop professional dialogue, it is hoped that a personalised approach to the school's curriculum reform can be developed. The following section of the literature review will focus on how lesson study or model lessons can be utilised to develop a collaborative approach to CPD in schools.

2.5.1 Lesson study as a form of CPD

Teachers can share knowledge and best practices in schools through lesson study or by observing model lessons. Lesson study, initially developed in Japan as a form of action research for professional development, can effectively improve teaching and learning (Kim et al., 2021), whereby teachers collaboratively plan, observe and analyse lesson sequences in the classroom.

Kim (2021) discusses lesson study as a genre of educational research grounded in classroom practice that in-service and pre-service teachers can use for professional development. Teachers collaboratively design a lesson, which one teacher then teaches whilst other school members observe and jointly discuss improvements (Kim, 2021). While initially developed for mathematics teaching, it has been applied to other curriculum areas and could be used to allow teachers to develop their computing subject knowledge collaboratively. Still, it would be most successful where more than one teacher taught the same year group. McNeill et al. (2016) also argue that the opportunity to observe peers and to be observed can be an integral part of successful CPD; they do, however, note that this needs to take place in a secure and trusting relationship, as many teachers' experience of observation is linked to stressful situations such as OFSTED inspections or performance management.

While research analysing the impact of lesson study on teaching computing in UK primary schools is relatively rare, studies in other subjects, such as mathematics (Vermunt et al., 2023) and the use of technology (Wang et al., 2024) can be usefully applied. In a large-scale, London-based study conducted over two years, Vermunt et al. (2023) worked with 214 Primary and secondary mathematics teachers from 59 schools to investigate the impact of lesson study on student and teacher learning. Their research suggests that lesson study can provide a mechanism for addressing practice problems, such as the need to introduce curriculum changes, and that these changes can be developed collaboratively and positively influence school culture. This was especially the case when school leadership were involved in and supportive of the process.

A few studies suggest that recording lessons can be beneficial in developing professional development through lesson study. Wang et al. (2024) found that teachers can struggle to identify key information in peer feedback when developing TPACK through a lesson-study approach. Working with 26 primary school language teachers in China, they designed a visualisation-based dashboard in which teachers uploaded videos of their teaching for peer feedback on their use of technology. This approach allowed teachers to obtain essential information from peer feedback, including suggestions for improvement, in a more supportive way. Although conducted in a different national and subject context, the study illustrates how video-based reflection and structured peer feedback can help teachers focus on specific aspects of practice, a principle that may be relevant when supporting English primary teachers to develop confidence with computing and CT.

An alternative approach to lesson study, typically carried out by peers, involves using model lessons to develop classroom teaching. Sims et al. (2023) used video modelling of questioning for retrieval to enhance pre-service teachers' practice in England. Working with 89 London-based trainee teachers, students observed a simulated classroom in which a teacher consistently demonstrated questioning for retrieval. Their study found that observing a model video had a positive impact on trainee teachers' questioning skills, compared with when students had read relevant research. All students who participated in the study demonstrated an increase in self-efficacy. This suggests that observing a model lesson may be a valid approach to CPD; however, because this study took place in a classroom simulator rather than in a classroom, it would need to be tested in a school setting to

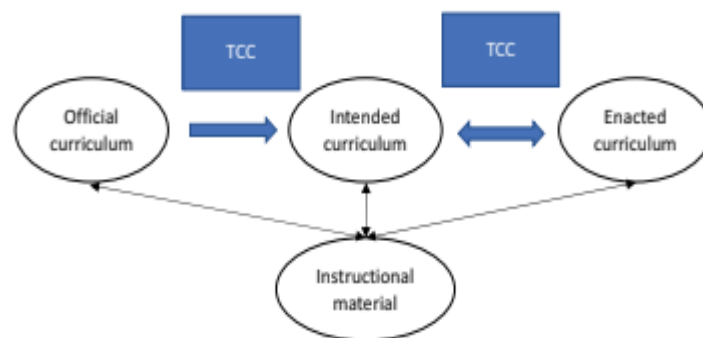
determine whether this approach is entirely replicable. The participants were trainee teachers rather than in-service teachers, so it will be interesting to see if this approach works with qualified teachers, suggesting an area for future research.

2.6 Curriculum design in schools

Alongside CPD to develop teachers' subject knowledge, it is important to focus on how the curriculum is planned and enacted in a school. The English National Curriculum for Computing (DfE, 2013c) describes the knowledge that pupils should know by the time they leave primary education; however, how this is enacted in schools and the pedagogy, or process of learning and teaching used when teaching it (Palaiologou & Kingdon, 2022), has a significant impact on what children learn. Deng (2022) argues that teachers should be viewed as curriculum makers through their interpretation of the National Curriculum. At the classroom level, teachers interpret and transform the institutional curriculum into instructional events, creating the enacted curriculum.

Figure 2.3

Teacher Curriculum Competence (TCC) (Tran and O'Connor, 2024, p. 8).



The Teacher Curriculum Competency (TCC) model (see Figure 2.3 above) describes how teachers draw on their knowledge of students' needs and attainment, as well as various resources and materials, including official government and school curricula, to create an intended curriculum (Tran & O'Connor, 2024). This may be expressed in lesson plans, but much of the information may not be recorded. Tran and O'Connor's (2024) conceptual model argues that TCC occurs as teachers transition from the official to the intended curriculum, utilising relevant instructional materials, and again as teachers transition from the intended to the enacted curriculum. They argue that TCC can also be reflective, as seen in the reverse arrow in diagram 2.2, when teachers' enactment informs future teaching, with knowledge informing the future intended curriculum. If, as Tran and O'Connor (2024) argue in their model, teachers' knowledge and thoughts about the importance of a subject, along with their position as a gatekeeper between the official curriculum and what is planned and enacted in schools, means that involving teachers in curriculum design is key to the successful development of a subject in schools.

The twin goals of the current study, changing the curriculum at The Watermill School and improving its teaching through quality Continuing Professional Development (CPD), may be achieved through the collaborative design (co-design) of the curriculum. Voogt et al. (2015) describe co-design as the process where 'teachers create or adapt existing curricular materials in teams to comply with the intentions of the curriculum designers and with the realities of their context' (p. 260). This process often involves external experts who provide insight into the rationale for change, and the collaborative

process offers opportunities for reflection on reform. Voogt et al. (2015) argue that involving teachers in collaborative curriculum design leads to greater change, as teachers are more likely to assume both individual and collective responsibility, resulting in transformative action and learning.

Scornavacco, Kelly and Boardman (2022) using qualitative data from a multi-year co-design research project in the USA, explored how co-design aligned with and promoted teacher leadership. Their research concluded that co-design supported teacher leadership by fostering ownership of innovation, enabling teachers to decide how curricula are developed authentically. Importantly, co-design also supported the development of a ‘cultivation of a shared vision for teaching and learning’ (p. 352), strengthening teachers’ collective commitment to the changes being introduced. These findings are highly relevant to the English primary setting; working collaboratively with a middle leader to redesign the computing curriculum is likely to support change more effectively, as teachers are generally more inclined to implement curricular approaches they have helped to shape. The following section considers how this collaborative approach can be enacted more effectively in practice.

2.7 Working with teachers or middle leaders to bring about change in schools.

Grootenboer et al. (2020) argue that effective educational development needs to be school-based and driven by middle leaders in response to local needs and conditions. This must be “classroom-out” reform rather than “top-down” imposed change (p. 47). Whilst it is impossible to change the National

Curriculum (DofE, 2013), working with the school's computing subject coordinator, based on the school's needs, can alter how the curriculum is enacted in school, leading to classroom reform and longer-term change.

In addition to developing CPD to enhance teachers' subject knowledge, it is essential to work with the whole school to implement changes in how the subject is taught, how the curriculum is planned within the school, and how this is enacted. As described in the research context (Section 1.4), the school has strong leadership. Grootenboer et al. (2020) argue that, due to the complex and often messy nature of school-based educational development, middle leaders are crucial when facilitating curriculum and professional development, 'because they work in the hectic milieu in and around classrooms and can appreciate the convoluted nature of educational practice and change' (p. 116). Therefore, this section of the literature review will examine best practices for working with middle or teacher leaders in a school setting. These can be described as teachers with an acknowledged position of leadership within a school, such as for a subject or curriculum area, but who also have a significant classroom-based teaching role (Grootenboer et al., 2015; Wenner & Campbell, 2017).

According to Harris and Jones' (2019) editorial, three key dimensions of teacher leadership can lead to educational change through shared practice: influence over others, action beyond formally assigned classroom roles, and the development of pedagogical excellence within the classroom and beyond. One crucial way to achieve this is through subject leadership, as the subject leader can influence what happens in both their classroom and the school. Using the term "middle leaders," Grootenboer et al. (2015) argue that the role

of the subject leader is vital as a buffer between school leaders and teachers. By participating in both leadership and teaching practices within the school, they can collaborate with teachers to implement curriculum changes in their day-to-day practice.

The subject leader's role in the school means that they have significant influence over the planned curriculum and are also actively involved in teaching it. It is essential to recognise this, as Lipscombe et al.(2020) note school teacher leaders have a unique position when leading school-based initiatives. They will often be the instigator and drivers for the change, but as teachers, they are involved as co-learners and participants. A subject leader will, therefore, be involved in driving change in the computing curriculum, but will also learn new concepts alongside the rest of the teachers in the school.

Whilst there is little literature on the role of computing subject or middle leaders in schools, research on other STEM subjects can be helpful. Vale et al. (2023) examined the goals and activities of 196 mainly (71%) primary school middle leaders of mathematics in Australia. In common with England, they found that 42% of primary maths leaders received less than 2 hours a week for completing leadership activities; this was often related to the size of the school and how their responsibilities were perceived. Their study found that whilst almost half of their participants had goals related to improving teaching and learning, most management time was spent on activities such as purchasing mathematics materials. However, teachers also collaborated with other teachers to enhance their knowledge of the curriculum and their capacities and mindsets related to mathematics; this was often achieved through co-teaching and shared planning, aimed at improving pupils'

mathematics learning by enhancing teaching. Although this study took place in Australia, similarities in the education system mean that these goals are comparable to those found in English schools.

In their book for middle leaders in schools, Grootenboer et al. (2020, p. 109) discuss how middle leadership can be seen as 'practice-changing-practice' with the overarching aim of leading to school-based professional and curriculum development. They propose an *action-oriented professional learning* approach to educational change, which involves collaborative work focused on local issues and slow, systematic improvement to achieve sustainable progress. This approach reflects on evidence from practice and acknowledges the personal and emotional elements of educational change. This collaborative approach, which explicitly examines practice within the school, whether with or without external agency involvement, is well-suited to the current thesis. Working with the school's computing subject leader, who is passionate about computing education, can lead to a shared goal of improving the teaching of computing in the school, resulting in lasting changes to how the curriculum is perceived and taught.

2.8 Summary

This chapter has reviewed the literature on teaching computational thinking (CT) in schools. It first investigated why CT should be taught, examining the reasons for and against its inclusion in the English National Curriculum (DofE, 2013a), and then explored how and when it is best taught. Arguing for the introduction of Computational Thinking (CT) in primary school, where children are introduced to key concepts early, the reviewed

literature also advocates a developmental approach to teaching computing and CT, with concepts being introduced in a practical manner. A cross-curricular approach to teaching, linking key concepts to other subjects, can also be beneficial in a time-pressured primary curriculum.

Next, this review explored teachers' perceptions of teaching CT, finding that the inclusion of computing in the primary curriculum led to numerous challenges, with many teachers lacking confidence in teaching it and some choosing not to teach it. It then moved on to discuss factors that can prevent the teaching of computing in schools, with outdated technology, time and money highlighted as concerns to many, whilst other literature focused on teachers' beliefs regarding the usefulness of teaching computing, finding that teachers' self-efficacy regarding whether they believe they can teach computing can have an impact on whether or not it is taught.

Next, in discussing how to teach Computational Thinking to teachers, the reviewed literature showed that teachers preferred CPD conducted in their schools and tailored to the specific needs of the children they taught. CPD, which used collaborative professional enquiry, was found to be the most successful. CPD needs to focus on improving teachers' subject knowledge and confidence in teaching, while also ensuring it is replicable in the event of staff changes.

Finally, this chapter explored working with subject or middle leaders within a school, finding that strong teacher leadership significantly impacts how a subject is taught and is a vital part of developing and teaching a new curriculum, as well as Continuing Professional Development, that has a lasting

impact on teaching and learning. Curriculum co-design, working with teachers in a school to develop a curriculum relevant to all learners, is most likely to lead to long-term change in computing teaching.

2.9 Research Aims

Therefore, to enact change and improve the teaching of computing in a primary school, the overarching research question (MRQ) derived from the literature is as follows:

How can the teaching of computing for primary school children be improved in an English Primary school?

To answer this question, it is necessary first to consider what types of activities and approaches are already being used in the school by answering the following research question:

- Research Question 1 (**RQ1**) How is computing currently taught in the school? What are the key challenges?

The answers to this question will lead to further questions

Research Question 2 (**RQ2**): What approaches can support the school in overcoming these challenges?

Research Question 3 (**RQ3**): How should these approaches be designed to be effective in this school?

Research Question 4 (**RQ4**): How do these designed solutions impact participants' experiences and outcomes?

CHAPTER 3: Methodology and Methods

3.1 Introduction

This chapter of the thesis discusses how research was conducted first to understand the impact of a new computing curriculum on a one-form entry primary school, and second to implement a change in practice within the school. The research aim was pragmatic and followed a design-based research methodology. The chapter, therefore, begins by describing design-based research and a pragmatic theoretical perspective. As design-based research is iterative, the two separate phases of the research are defined with an overview of the materials collected. This is followed by a description of how the sample was chosen, based on the specific details provided in the introduction. Next, a description of the materials used to generate qualitative data is provided, followed by details of the data analysis methods employed. The chapter concludes with a consideration of the validity, reliability, and ethical issues pertinent to the research.

3.2 Research Aims

This research originated from informal conversations with students returning from teaching practice at the university where I work; students were saying that computing was not being taught or deliberately allocated to them as students to teach. If this were the case, it would mean that, despite introducing a new curriculum, school children were not learning a key National Curriculum subject (DfE, 2013b). Having identified a problem, I began exploring the literature on computing in schools. I noticed patterns related to teachers' lack of subject knowledge and, consequently, a lack of

confidence, which impacted whether computing was being taught. This study aimed to investigate whether this was the case in a small English primary school and to explore options to support teachers in teaching computing. This was addressed through four research questions:

RQ1: How is computing currently taught in the school? What are the key challenges?

RQ2: What approaches can support the school in overcoming these challenges?

RQ3: How should these approaches be designed to be effective in this school?

RQ4: How do these designed solutions impact participants' experiences and outcomes?

3.3 Design-Based Research

The proposed methodological framework for the research employed design-based research (DBR), a method for studying learning in environments such as schools that are designed and systematically modified by the researcher (Barab, 2006). DBR can be seen as a collection of approaches that 'involve a commitment to studying activity in naturalistic settings' to advance theory whilst directly impacting practice (Barab, 2014, p.151). It is not a single or static methodology, but can draw on methods from multiple traditions (Campanella & Penuel, 2021). As such, DBR moves beyond observing what is happening in a context to systematically change the context, enabling improvement and the generation of evidence-based claims about learning (Barab & Squire, 2004). DBR can be seen as a combination of

instructional design and educational research, often involving teachers from the outset (Prediger et al., 2015). DBR is pragmatic, seeking to impact practice directly, and may result in practical tools such as professional development designs and curriculum materials (Campanella & Penuel, 2021). However, these may not be widely shared.

Design-Based Research (DBR) was selected as the overarching methodological approach for this study because it enables researchers to design, implement and refine interventions in authentic school settings while generating both practical and theoretical insight (Bakker, 2018). This fits with the aims of the current study, which investigated the current practice of a one-form entry primary school in the first phase of the research and developed Continuing Professional Development (CPD) materials during the second phase to improve computing teaching. Unlike experiments that aim for statistical generalisability, DBR aims to generalise through the development of a design argument. By attending to the features of the design and the context in which it is implemented, it is hoped that findings can be generalised to other similar situations (Barab, 2022). DBR was chosen because, unlike other forms of educational research, it ‘provides a direct link between research and practice,’ increasing the chances of meaningful impact (Reeves, McKenney, and Herrington, 2011, p. 58). Alternative methodologies such as action research or case study designs were considered. However, action research places the practitioner at the centre of the enquiry (McNiff, 2017), whereas this research required a distinction between the researcher and practitioner roles as the researcher was not situated within the school. Similarly, while case study methods (Yin, 2018) support in-depth exploration of a context, they do

not inherently incorporate the design-implementation-reflection cycles (Bakker, 2018) central to curriculum innovation. DBR was therefore most appropriate because it uniquely integrates empirical investigation with iterative curriculum development, aligning directly with the aims of understanding and improving Computational Thinking in the school.

The research context, a one-form entry primary school in the North-West, is an important issue in DBR, as it recognises and focuses on the ‘messiness’ of real-world practice. The context becomes a core part of the picture, not ‘an extraneous variable to be trivialised’ (Barab & Squire, 2004, p.3). In addition, the study's participants, in this case, teachers within the school, were treated as co-participants in both the study's design and the analysis. The teachers’ knowledge of the context, specifically that of the computing coordinator, was utilised to bring their differing expertise into producing and analysing the design (Barab & Squire, 2004). In this study, the collaboration with the computing subject leader reflects what Gomez et al. (2018) describe as participatory co-design (PCD), where teachers and researchers jointly shape the instructional tools and pedagogical approaches used in the classroom. PCD assumes a ‘distributed expertise’ stance in which teachers contribute contextual, pedagogical and curricular knowledge, while researchers bring theoretical and design-oriented expertise (Kyza & Nicolaidou, 2017, p. 267). This alignment makes DBR particularly appropriate for addressing the study’s research questions, which focus on both understanding existing CT practice and building a viable intervention tailored to the local school context.

Another key factor of DBR is that the educational ideas for teacher and student learning are formulated in the design, but can be adjusted during the empirical testing, for example, if the design idea proves unsuccessful in testing. Design implementation often requires decisions beyond the scope of the design; no design can specify all the required details. The combination of researchers and co-participants may also lead to design changes; participants' actions during implementation will require constant adjustments and decisions about how to proceed at every level (Collins et al., 2004). This was relevant to the current research, as whilst the first phase of the design was fully formed, the second phase depended on the outcomes of the first phase. Designs in education can never be fully specified; design evaluation is ongoing and context-specific. The participants' needs, interests, abilities, interpretations, interactions and goals will significantly impact how the research proceeds (Collins et al., 2004). A close collaboration of the school practitioners and the researcher in DBR is required for testing and refinement of both the 'prototype solutions' and the design, meaning that educational design research is not completed until results represent progress in solving problems, in this case, the teaching of Computational Thinking, are attained (Reeves et al., 2011). Gomez et al. (2018) describe co-design as requiring trust, shared ownership and an iterative approach in which teachers 'try out' designs in real classrooms and collectively reflect on outcomes. This mirrors the collaborative cycles undertaken in the present study with the computing lead, in which planning, enactment, and refinement occurred iteratively, grounded in the practical realities of the school. Such collaboration ensures that any resulting curriculum materials or pedagogical guidance are not only theoretically

informed but also feasible and meaningful in the English primary context. The second phase of the research, which examined the implementation of a change to primary computing teaching in a specific school, was developed in response to the picture presented in phase one.

The phases of the research project can be viewed as layered; the first phase aimed to describe and explain the existing teaching of computing within the one-form entry primary school. Whilst some instances of good practice were seen within the school, the second advisory phase sought to develop and improve the teaching of computing (Bakker & van Eerde, 2015). Good practice instances were used to inform the research and the CPD offered to staff within the school. DBR can therefore be seen as an interventionist approach that seeks to effect change by intervening in what is happening in the school and classroom. It can also be seen as iterative, with a cycle of design, implementation, analysis and redesign (The Design-Based Research Collective, 2003).

3.4 Theoretical perspective

In addition to refining practice, design-based research must address theoretical questions and issues to be effective (Collins et al., 2004). DBR can be seen as being ‘largely agnostic’ regarding the methodologies used in research, typically involving mixed methods that utilise a range of research tools and techniques (Anderson & Shattuck, 2012, p. 17). The epistemological assumptions will, therefore, include the idea that the ‘traditional dualism’ between quantitative and qualitative research is not helpful, so that ‘good research should focus on the possibilities of integrating quantitative and

qualitative approaches rather than keeping them separate' (Denscombe, 2010, p. 129).

3.4.1 Pragmatism

The primary philosophy guiding the current research is pragmatism, which posits that knowledge is acquired through a combination of action and reflection (Biesta, 2010). A pragmatic perspective rejects the traditional paradigms of 'realist vs. idealist ontology' and 'subjective vs. objective epistemology' (Coe et al., 2017, p. 8), seeing them as problematic and unhelpful. A pragmatic approach focuses on achieving the best outcome from research, arguing that what is important or valid is what helps solve problems; research design should, therefore, be planned and conducted in the best way to answer the research question (Johnson & Christensen, 2024). Biesta (2010) argues that pragmatism should be seen as a set of tools for addressing problems rather than a philosophical position. Therefore, pragmatism's ontological assumptions are inconsistent with a specific philosophical position; it looks for the practical usefulness of the outcome. This is consistent with the research's aim to explore the current state of computing teaching in a specific school and implement change.

The pragmatic approach to the current research allowed the research questions to be answered regardless of philosophical presuppositions, redirecting attention to 'methodological rather than metaphysical concerns' (Robson, 2011, p.171). The research was based on the assumption that 'collecting diverse types of data best provides a more complete understanding of (the) research problem than quantitative or qualitative data alone' (Creswell, 2014, p.19). The current research, therefore, employed a mixed-

methods approach to investigate the quality and quantity of computing teaching in a one-form-entry primary school in North West England. Mixed-methods approaches allow for ‘what’ and ‘how or why’ research questions to be answered (Cohen et al., 2011).

3.5 Research Design

Design-based research can be seen as an approach to researching activity in naturalistic settings rather than an explicit method (Barab, 2006). The mixed-methods approach to the research enabled a comprehensive picture of computing within the school to be gathered. Methods used during the different phases of the research varied as new needs and issues emerged, and the research focus subsequently evolved (Wang & Hannafin, 2005). As such, the design approach needed to be interactive, flexible, and iterative.

3.5.1 Phase one – What does computing currently look like in a one-form entry primary school in North West England

The first phase of the current research aimed to obtain a rich picture of the current state of computing within the school to answer RQ1:

How is computing currently taught in the school? What are the key challenges?

A range of data was collected from various sources to provide a rich depiction of the school's context and to identify all the characteristics of the school or variables that may be influencing computing teaching. The school was a one-form entry school, with seven classes, organised by academic year group for children aged 4-11, and a maximum of 210 pupils. Each class was taught by a class teacher, with additional teachers providing class cover as

required (see Section 1.3.2 for further details). All the teachers agreed to some extent to be involved in the study. The school also had teaching assistants who worked with children across the school; some of these teaching assistants, who worked in Key Stage 2, were involved in the study either through interviews or by being present during lesson observations or model lessons.

The first phase of the research was an exploratory phase that aimed to determine the status quo in the school prior to the intervention (Robson, 2011). Semi-structured interviews were conducted with the head teacher and computing coordinator. Additionally, document analysis was performed on freely available curriculum documents, and a learning walk was conducted with the computing coordinator, observing two teachers teaching computing and seeking evidence of computing on display in the school and the children's work. Finally, four key stage two teachers and two teaching assistants were interviewed to discuss their feelings about the computing curriculum and what contributes to it. At the same time, field notes were taken during meetings with the computing coordinator as a co-participant, and during a computing lesson taught in the Year 6 (age 10-11) class. Table 3-1 illustrates the data collected during Phase 1.

Table 3.1*Data Collected During Phase 1*

Data generation activity	Amount of data collected	Participant/s	Collection date	Content covered	Analysis
Semi-structured interview	1x 1-hour interview	Headteacher and computing co-ordinator	Jan 2018	How has the new computing curriculum impacted the school's curriculum? What concerns do they have regarding computing? What is affecting the amount of computing that gets taught?	Summary Transcription Coding of themes
Document analysis of the original school computing curriculum from 2017/18	1 x written document of the planned curriculum	Computing coordinator for use by the rest of the staff	Jan 2018	The planned version of the computing curriculum for the Watermill School* How does the curriculum compare to the published National Curriculum?	Coding of themes generated from the National Curriculum
Lesson observations (Phase 1)	2 x 30-minute observations		June 2018	Learning walk of the school, looking for evidence of computing (displays, etc) with Alex, including observations of two members of key stage 2 teaching computing, discussions related to	Descriptive observation and field notes Transcription of audio recordings Coding of themes

Teacher-written interviews	2 x short written interviews	Year 5 teacher James * ⁶ Year 3 teacher Rose * (now retired).		the content covered and responses to written interviews	
Meeting with co-participant	1 x 1-hour meeting	Computing co-ordinator (Alex*)	June 2018	Discussion of observations and what is happening in the school regarding computing. Discussion of curriculum and next steps	Summary Transcription Coding of themes
Computing teaching Hannah	1 x 90-minute lesson	Hannah* teaching year 6 (for PPA)	June 2018	Transition lesson for year 6. Researcher teaching year 6 using university micro:bits	Field notes Descriptive observation
Semi-structured interviews	4 x 30-minute interviews	Teaching staff Matt* Year 2 Chloe* Year 3 Hannah* Year 4 James* Year 5	December 2018	Teachers' views on the computing curriculum: Frequency of teaching computing When is computing taught and how often? What has been taught? What works/ what does not work? Basic Feelings around teaching computing What do they feel computing is?	Transcription Coding of themes

⁶ All names of both the school and participants are pseudonyms

				Skills level– what are they struggling with?	
Semi-structured interviews	2 x 20-minute interviews	Year 2 Teaching Assistant Year 4 Teaching Assistant	January 2019	TAs' views about computing/ subject knowledge/and confidence	Transcription Coding of themes

3.5.2 Phase 2- Developing Computational Thinking within a one-form entry school

Phase 2 of the research was designed in direct response to the analysis of the data collected in Phase 1 and informed by the literature on Computational Thinking and continuing professional development. Following a discussion of the school's planned and enacted curriculum with the computing coordinator, he rewrote the curriculum to align with the National Curriculum objectives (DfE, 2013c). Further document analysis was then carried out of the new planned curriculum and teachers' medium-term planning (termly block overviews).

A lesson-study approach was employed as continuing professional development (CPD) with the computing coordinator (Alex), who acted as a change agent and co-participant in the study. First, I taught model lessons to the Year 5 class before we carried out shared planning and team-teaching; subsequently, Alex took over planning and teaching lessons. Audio recordings of Alex teaching were transcribed and analysed.

Finally, CPD for the rest of the school's teachers was designed, drawing on the literature and responding to the school's specific context. This is described in Chapter 4. The CPD sessions were followed up with further meetings with staff and the development of model lessons.

Before the COVID-19 pandemic, the original aim of the study was to return to the school in the summer term, after teachers had worked with the curriculum for six months, to reinterview teachers and observe computing

lessons. The pandemic, including several lockdowns during which most children were educated at home, meant that computing was not a priority for teachers who had to rapidly adapt to online learning and plan lessons for children to participate in from their homes. When children were being educated in the classroom, COVID-19 distancing procedures meant that external visitors were not allowed in schools. This meant that the empirical phase of the research ended sooner than initially planned, and the study design had to be altered. Table 3.2 illustrates the data collected during phase 2.

Table 3.2*Data Collected during Phase 2*

Data generation activity	Amount of data generated	Participant/s	Collection date	Content covered	Analysis
Document analysis of the rewritten school computing curriculum from 2018/2019	1 x written document of the planned curriculum	Computing coordinator for use by the rest of the staff	Sept 2018	The planned version of the computing curriculum for the Watermill School* ⁷ How does the curriculum compare to the published National Curriculum?	Coding of themes generated from the National Curriculum
Document analysis of school block overviews (Published on the school's website and	14 x curriculum overview documents	All teaching staff	Throughout the 18/19 academic year	How the computing curriculum is recorded in the teacher's medium-term planning. Evidence of changes to the amount of CT content planned	Coding of themes generated from the National Curriculum

⁷ All names of both the school and participants are pseudonyms

given to
parents as a
guide to the
planned
medium-
term
curriculum
for the half
term).

Meeting with computing co-ordinator	1 x 1-hour meeting	Alex*	October 2018	Discussion around the computing curriculum at The Watermill School, examining how to integrate Computational Thinking into the curriculum and professional development.	Field Notes Transcription of an audio recording. Coding of themes
Lesson/ participant observations	1 x 1-hour lesson taught by Researcher	Alex* teaching year 5 for James’* PPA	November 2018 – May 2019 (weekly)	First session: Researcher teaches Introduction to the micro:bits to Year 5 while AL observes. Discuss what the sequence of lessons would look like. Session 2-4: Joint planning of session and team teaching with Researcher Session 5: AL plans and teaches the session, Researcher observes	Field notes Audio recording of the final lesson Transcription of an audio recording Coding of themes
Continuing professional development	3 x 1-hour lesson co- taught 1 x 1-hour lesson observation of Alex*				

Meeting with computing co-ordinator	1 x 30-minute meeting	Alex* as computing co-ordinator	May 2019	Discussion around next steps/ CPD	Field notes
Continuing Professional Development	1 x 1-hour session plus follow-up	Teaching staff	September 2019	CPD session for all teaching staff at The Watermill School*	Field notes
Document analysis of artefacts from CPD	6 x curriculum maps annotated during the session	Teaching staff	September 2019	Emails from staff post-CPD	Document analysis
Curriculum design meeting	1 x 30-minute meeting	Meeting with Sally* Year 6	October 2019	Planning year 6 cross-curricular Computational Thinking based on the topic.	Field notes
Model teaching CPD lesson follow-up	1 x 1-hour lesson	James* Year 5	October 2019	Teaching micro:bits session to year 5 with James* observing. James* was then tasked with teaching follow-up lessons.	Field notes

Table 3.1 Data collected during Phase 2

3.6 Sampling

The study used a non-probability purposive sample (Cohen et al., 2011) which did not set out to represent the broader population of schools. One school was chosen as an instrumental case (Candappa, 2017) examined to provide an insight into the picture of computing within a school, providing a localised impact. There will be no attempt to generalise from the school to schools in general; however, a rich description of the context is provided so that others can make connections to their contexts (Barab, 2006) and the advancement of theory leading to theoretical generalisation (Robson, 2011).

The school was chosen based on its possession of the characteristics sought and as a typical example of a one-form entry primary school in the North-West. As the study involved participatory research, with the computing coordinator involved in the design process, the school was also selected due to the need for a computing coordinator who was interested in the subject and willing to co-design resources.

Access to the school involved gaining formal agreement from the head teacher of the school, acting as a gatekeeper (Cohen et al., 2011) gained through connections (Robson, 2011) formed through working as a primary education computing lecturer at a university in the North West.

3.7 Materials and Data Analysis

Evidence was collected from a broad data corpus (Braun & Clarke, 2006) in order to gather a rich picture of computing teaching at The Watermill

School. A range of materials was therefore used; the complete list of data gathered, along with the materials used, is included in Section 3.5.

3.7.1 Semi-structured interviews

Data was collected through face-to-face semi-structured interviews (Robson, 2011) with the head teacher, computing coordinator, teachers at the school, and teaching assistants. In total, seven semi-structured interviews were conducted during phase one, combined with data from two written interviews, with the staff listed in Table 3.3 below (pseudonyms have been used for confidentiality).

Table 3.3

Semi-structured interviews conducted in Phase 1

Interviewee (Pseudonym)	Responsibility
Sandra	• Headteacher • Non-teaching management role
Alex	• Computing Lead • KS1/KS2 • Teaches across different classes
Matt	• KS1 Teacher • Newly Qualified Teacher
Chloe	• KS2 Teacher • Recently Qualified Teacher
Hannah	• KS2 Teacher • Experienced Teacher
James	• KS2 Teacher • Experienced Teacher • Also completed a written interview
Rose	• KS2 Teacher • Experienced Teacher • Written interview only
Amy	• KS1 Teaching Assistant
Louise	• KS2 Teaching Assistant

One teacher, Caroline, did not wish to be interviewed as part of the study, and another, Sally, joined the school during Phase 2 and was therefore excluded from the interviews. Further information about the participants is provided in Section 1.3.2, which provides a more detailed examination of the school's context.

Semi-structured interview questions (see Appendix 1 and 2) were used as a guide or checklist of topics to cover, with set wording and a predetermined order for the questions (Robson, 2011). Semi-structured interviews allow for questions to be modified based on the interview flow, with additional questions added to follow up on responses. Open-ended questions were asked about computing within the school to elicit views and feelings regarding computing. The teachers' interview questions aimed to gather data about the amount of computing teaching that occurred, who usually taught computing, how teachers felt about teaching computing, and what contributed to this, as well as any barriers to teaching computing and the teachers' past experiences with learning computing. Those teachers who had taught under the previous curriculum were also asked to consider how the two curricula compared. Interview questions for the headteacher focused on teaching across the school and aimed to elicit responses related to challenges to implementing the new curriculum, as well as how teachers had responded to the change. The computing coordinator was also asked about challenges of changing the curriculum, the role of the computing coordinator in the school, and how this role may have changed, as well as questions designed to gather information about resources at the school.

The interviews were all conducted face-to-face and recorded using a digital Dictaphone before being manually transcribed. The teacher and TA interviews were conducted in one-to-one settings; however, due to staff availability, the interview with the head teacher and the first interview with the computing coordinator were conducted in a triad, which may have impacted some of the responses.

3.7.2 Publicly Available Documentation from The Watermill School

As part of the research, several publicly available ‘found texts’ (Thompson & Hall, 2017) were used. These included the school’s original published planned computing curriculum, the revised computing curriculum, and teachers’ block overviews or medium-term planning.

3.7.3 Observations and Field Notes

Participant observations (Robson, 2011) were conducted to document the participants’ experiences in teaching computing lessons. A learning walk was conducted with the computing coordinator in the afternoon, where all teachers in the school were asked to teach computing. Two teachers (Rose and James) agreed to be explicitly observed teaching computing lessons during which an audio recording was made using a digital Dictaphone, and field notes were taken of informal discussions.

The material environment of the school was also observed (Thompson & Hall, 2017), with a focus on identifying evidence of computing in displays, documenting where computing was taught, and noting the location of computers, interactive whiteboards, equipment used and teachers’ use of technology. Notes from these observations were reviewed immediately

following each visit to the school and formed the basis of follow-up meetings and discussions with the participant teacher.

3.8 Data Analysis

Cohen, Manion, and Morrison (2018) suggest that there is no right or wrong way to analyse and present qualitative data; the method used should be chosen for 'fitness for purpose', with the primary purpose of the analysis being to turn data into findings. They further suggest that qualitative data analysis relies on interpretation, with multiple possible interpretations. There is often a merging of analysis and interpretation, with data analysis occurring while data are collected. In this study, the range of data collected, including spoken and written interviews, lesson observations, field notes, and documents, required searching across the dataset to find repeated patterns of meaning or themes (Braun & Clarke, 2006). Braun and Clarke's (2006) six stages of thematic analysis were used in analysing the data. Interviews with the co-participant, headteacher, teachers in the school, and teaching assistants conducted during Phase One were recorded, transcribed, coded, and analysed using an iterative coding scheme and a mixture of inductive and deductive codes.

All interviews were transcribed by the researcher, including a summary of the interview content (Rubin & Rubin, 2005). The differences between oral speech and written text mean that transcription is an interpretative process; whilst the interview can be seen as an evolving conversation between two or more people, when transcribed, it becomes 'abstracted and fixed in written form' (Brinkman and Kvale, 2015, p. 203). As the content of the interview, rather than the organisation of the language, was of more interest to the

research questions, a simplified form of transcription was used; due to the nature of the interviews, it was not felt necessary to record interactions such as pauses or inflexions (Flick & Metzler, 2014). The transcripts can, therefore, be seen as a means or tool for interpreting what was said during the interview (Brinkmann & Kvale, 2015). In this instance, it was essential to recognise that interviews are not transcripts but rather should be viewed as living conversations.

Therefore, a narrative approach was taken in the interview analysis, which builds upon the original story told by the interviewee and anticipates the final story to be reported to the audience. As such, the interviewer can be seen as the ‘co-producer and co-author’ of the interview, with the interviewer's active listening and follow-up on answers co-determining the course of the interview (Brinkmann & Kvale, 2015, p. 219). The transcripts were analysed to identify concepts, themes, and events related to current issues in computing within the school. Coding was used to label the different concepts and themes identified across the separate interviews and open-ended questionnaires. Coding was used as the primary method of qualitative data analysis because it enables large volumes of text to be reduced to manageable, meaningful units while preserving the integrity of the original data. As Cohen et al. (2018, p. 1114) note, coding is a ‘major approach to qualitative data analysis’ that supports the systematic organisation of data into categories and themes, which involves breaking down text into smaller units, comparing and conceptualising these units and assigning labels that reflect their meaning. Coding, therefore, provided a transparent and replicable structure for analysing interviews,

observation notes, and documents, enabling the identification of patterns across the data (Cohen et al., 2018).

Data from each teacher observation and interview were coded separately, with each teacher viewed as a case. Individual cases were then analysed in a cross-case analysis, which identified similarities and differences across the cases (Israel et al., 2015). The codes were continually checked to ensure no drift in their meaning, thereby maintaining reliability (Creswell, 2014). Items that have been allocated the same code were combined in separate files (Rubin & Rubin, 2005).

Profiles or vignettes of a participant's experience were crafted as an effective way to share interview data (Seidman, 1998). These were written in the first person, using the interviewees' voice where possible; when necessary, the researcher added words to clarify and deleted 'characteristics of oral speech that a participant would not use in writing' (Seidman, 1998, p. 104) to represent the participant's voice authentically.

The study used a combination of deductive and inductive coding, in line with recommendations from Cohen et al. (2018). Three separate coding schemes were developed to answer different research questions.

An inductive coding scheme (Scheme A, Table 3.5) was used to answer the sub-question: *How is computing currently taught in the school?* This explored who was teaching computing, when it was taught, and how it was taught. In developing the inductive codes, the analysis followed Braun and Clarke's (2006) guidance on data-driven thematic analyses. Following repeated readings of the transcripts, the data were coded line-by-line to identify

meaningful features without imposing the deductive categories derived from Sentence and Csizmadia (2017). Through this iterative reading, recurring ideas and issues raised by teachers were noted and developed into inductive codes that captured aspects of their practice and understanding not covered by the deductive framework. Coding was carried out systematically across all transcripts, with equal attention given to each, and extracts were coded in more than one way where this supported a fuller representation of their meaning. This inductive strategy allowed codes to emerge from teachers' own words, concerns and practices, supporting the identification of unanticipated barriers, uncertainties and contextual factors that may not have been captured through deductive coding alone (Braun & Clarke, 2006).

Scheme B (Table 3.6) used a combination of deductive and inductive codes to address the sub-question: *What are the key challenges?* Deductive codes were derived from Sentence and Csizmadia (2017) Computational Thinking framework, providing a theory-driven set of categories aligned with the research questions. Where a suitable code was not available, inductive coding was used and added to the codebook (Table 3.6). As Cohen et al. (2018, p.1113) argue, researchers often begin with codes that come from 'theoretical constructs or areas of interest devised in advance of the analysis.' Alongside these, inductive (emergent) codes were generated in response to close reading of the data, reflecting meanings that arose naturally during analysis. This approach aligns with Cohen et al.'s (2018) description of coding as both deductive and inductive, allowing new insights to emerge from the data rather than forcing all data into predefined categories. Some of the deductive codes derived from Sentence and Csizmadia's (2017) framework were not present in

the data; this reflects Cohen et al.'s (2017) argument that coding frameworks derived theoretically do not map perfectly onto empirical data; coding requires iterative checking, refinement and comparison. These codes remain in the coding scene to maintain alignment with the original theoretical framework. The identified themes were sorted into three overarching areas, also drawn from Sentence and Csizmadia (2017).

In developing the inductive codes, the analysis followed Braun and Clarke's (2006) guidance on data-driven thematic analyses. Following repeated readings of the transcripts, the data were coded line-by-line to identify meaningful features without imposing the deductive categories derived from Sentence and Csizmadia (2017). Through this iterative reading, recurring ideas and issues raised by teachers were noted and developed into inductive codes that captured aspects of their practice and understanding not covered by the deductive framework. Coding was carried out systematically across all transcripts, with equal attention given to each, and extracts were coded in more than one way where this supported a fuller representation of their meaning. This inductive strategy allowed codes to emerge from teachers' own words, concerns and practices, supporting the identification of unanticipated barriers, uncertainties and contextual factors that may not have been captured through deductive coding alone (Braun & Clarke, 2006).

A deductive scheme (Scheme C, Table 3.4) using codes generated from the National Curriculum (DfE, 2013c) was used to analyse the planned school curricula and teachers' planning to examine what was being taught. Each National Curriculum objective was assigned a corresponding

code, and the curriculum documents were carefully read and re read to determine which curriculum objective was being taught.

The same three coding schemes were used in the analysis of Phase 2 to analyse teacher planning, observations of teaching and interviews with the co-participant. Additional inductive codes were developed through close reading of the data as needed to address challenges identified during Phase 2 observations of teaching. Due to the nature of the observations, resource-related challenges were divided into iPad- and micro:bit-related. Further inductive codes were also added to cover student behaviour and rushing.

Nvivo 12 (Jackson & Bazeley, 2019) was used to organise and manage the coding process. As Cohen et al. (2018, p. 1116) emphasise, computer-assisted data analysis software such as NVivo supports researchers in retrieving, categorising, collating, and separating data. Using Nvivo enabled consistent coding across the dataset, supported the refinement and modification of codes through iterative analysis and enhanced the dependability and transparency of the analytical process.(Jackson & Bazeley, 2019).

Table 3.2

Deductive Coding scheme (Scheme C) for document analysis of planning and curriculum documents based on The National Curriculum Computing Programme of Study (DofE, 2013b).

Research aims	Codes	Description	Example
What parts of the curriculum are being taught?	<u>Key stage 1</u>	<u>Computing Curriculum in Key Stage One</u>	
	Algorithms	Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions.	I can explain how you program a toy.
	Create and debug	Create and debug simple programs.	I can test a friend's instructions.
	Digital content	Use technology purposefully to create, organise, store, manipulate and retrieve digital content.	I can select sound clips to accompany an image.
	Logical reasoning 1	Use logical reasoning to predict the behaviour of simple programs.	I can predict what will happen with instructions.
	Safe technology	Use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.	I can explain that all information published on the internet leaves a trail that stays forever.

Uses of technology	Recognise common uses of information technology beyond school.	I can talk about how computers help to control machines and toys.
<u>Key stage 2</u>	<u>Computing Curriculum in Key Stage Two</u>	
Acceptable behaviour	Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.	I can explain why people use passwords and learn the benefits of a strong password.
Logical reasoning 2	Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.	I can use logic to predict and analyse what will happen when I have finished creating my algorithm.
Networks	Understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web; and the opportunities they offer for communication and collaboration.	I can understand and explain that the internet contains lots of information/data from different people.
Programs	Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.	I can plan a sequence of algorithms to control movement on a toy/programme.

Search	Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content.	I can compare data/information generated by a search engine and locate the most appropriate 'hit'.
Software	Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.	I can understand and explain how Microsoft PowerPoint works and how it can be used to present information in a formal and informal manner in a slideshow (PowerPoint).
Variables	Use sequence, selection, and repetition in programs; work with variables and various forms of input and output.	I can combine short sections of an algorithm to create a clear sequence.

Table 3.5*Inductive Coding Scheme for Phases 1 and 2 (Scheme A)*

Research aims	Codes	Description	Example
How much computing is taught?	Time	How much computing is being taught?	“So we do e-safety every week of 20 minutes.”
How is computing taught?	Approach	What approaches are used to teach computing?	“We've got the LEGO bricks bought as well, which we use well, we've used them well.”
Who is teaching computing?	Taught by	Who is teaching computing lessons?	“But I have Alex who teaches (the class) for a full day for me...”
When is teaching caught?	When taught	When is computing being taught?	“So it's normally on Monday afternoon”

Table 3.6

Deductive coding scheme (Scheme B) based on Sentance and Csizmadia (2017) with inductive codes (highlighted) developed from the data.

Research aims	Codes	Description	Example
What factors are affecting the teaching of computing within the school ⁸ ?	Choosing resources/ Finding quality resources	Not knowing what resources to use and the time spent sourcing resources	“I think you are always whenever you think you or the latest thing you
Challenges relating to resources	Lack of resources	Challenges are caused by either not having enough equipment or by not having the right equipment.	“Not having the equipment me, that’s a real bugbo
	Computing resource management	Challenges caused by allocating computing equipment to children.	The start of the lesson iPads to children; whi it took up time in the l
	Technical problems.	Challenges caused by outdated equipment or infrastructure.	“Each time we do a ne pieces that we didn't k I'm sure we've never u
	Students and instructions	Children's ability to follow instructions	Not present in the data

Challenges relating to students	Students and practice	Children needing/ or not needing additional practice of procedures.	“Reinforcement regular (the children) quite comfortable aren't they... they love
	Students and remembering	Children remember or do not remember information and procedures from prior learning.	“I’m always impressed children get used to whether I’m teaching
	Students and confidence	Children who have or lack the confidence to have a go at computing.	“Computing’s one of t enjoy and it’s one of t feel really quite confident make the effort to do.”
	Students and engagement	Children are engaged or not engaged in computing.	“I think they love it, th if it’s iPads everyone
	Students and understanding	Children understand or do not understand computing.	“I think now we’re in children know technology how to use it”
	Students and problem-solving	Children have or do not have the skills to solve or debug problems themselves.	“That’s kind of what y and actually it’s not w make mistakes”
	Students are starting too late.	Children have missed prior objectives due to the new curriculum's introduction or because the objectives were not taught.	“It was a surprise in th Early Years children c knowledge in terms of

	Students' use of language	Children did not have the language to explain what was happening.	where the early years change for them" "The thing I picked up particularly the year 3 well, and I was saying actually making?' and algorithm or coding"
Challenges relating to teachers	Teachers and coding focus	Teachers overfocused on coding.	"This seems slimmer
	Teachers and consistency	Teachers and the lack of consistency in teaching	"...having the correct making sure that we to throughout the year gr teaching it their taught versa if the teacher's t so just keeping that co quite a bit of a barrier
	Teachers and digital literacy	General confidence in using computers	"I love PowerPoint pr but even those are get other options, so they
	Teachers and dry topics	Teachers find the topics too dry.	Not present in the data
	Teachers and funding	Teachers and a lack of money	"So, as always with IT money and resources."

Teachers and a lack of training	Teachers lacking opportunity for professional development in computing.	“So just the training, to teach each topic really properly.”
Teachers and subject knowledge	Not having enough subject knowledge to teach topics confidently.	“My subject knowledge is great, (on) the specific topics.”
Teachers and troubleshooting	Teachers either know how to troubleshoot computing problems or do not.	Not present in the data.
Teachers' lack of confidence	Teachers are nervous about teaching computing.	“I don’t mind computing, but the resources, my confidence that sounds ridiculous.”
Teachers' lack of time	Teachers do not have enough time to research or teach computing topics.	“(we try and make it work, but we got enough time to plan.”)
Teachers' use of language	Teachers are unsure about the language associated with computing.	“definitely language, but confidence using the correct language.”
Teaching approaches	Different approaches to teaching computing	“The teacher's dialogic approach and is mainly didactic, giving directions (I.e. to swap roles) engaged in dialogic discussion, their thinking.”

3.9 Research Validity and Reliability

Bakker and van Eerde, (2015) identified both internal and external validity and reliability as the most relevant concerns for design-based research. Internal validity seeks to show that the data can sustain the explanation of the research (Cohen et al., 2011). External validity, however, refers to the extent to which results can be generalised to wider populations.

The context of the DBR research is therefore important to the validity of the research; theoretical claims need to be substantiated with rich and meaningful descriptions of the context (Bakker & van Eerde, 2015). Design-based researchers not only recognise the importance of local contexts but also treat changes to these contexts as necessary evidence for the validity of a theory (Barab & Squire, 2004, p. 6). Internal validity was therefore enhanced through gathering a rich collection of data; field notes, interview data and curriculum documents were collected to allow data triangulation to test conjectures, whilst the iterative nature of the research allows conjectures from earlier in the research to be tested in later iterations (Bakker & van Eerde, 2015). The rich descriptions gathered, specifically of the context, enable readers and users of the research to determine whether the findings are transferable to other contexts, such as other schools (Cohen et al., 2011), thereby impacting the external validity of the research.

Ecological validity, a form of external validity, can also be seen as important in education as it concerns examining and addressing the specific characteristics of a particular situation and therefore requires the specific

factors of the one-form entry primary school, as the research site, to be included and taken into account in the research (Cohen et al., 2011). For ecological validity to be demonstrated, it will be important to include ‘as many as possible of the characteristics and factors’ of the school (p. 195). This, however, can raise ethical concerns, as the more detailed the description of the environment, the more likely the school will be identifiable, thereby making non-traceability, anonymity, and non-identifiability harder. This is addressed further in the ethics section and during the ethical approval process.

Reliability can be seen as dependability, consistency and replicability over time, instruments and groups of respondents (Cohen et al., 2011). Internal and external reliability are important for design-based research. Internal reliability denotes ‘how independently of the researcher the data are collected and analysed’ (Bakker & van Eerde, 2015, p.25). This was improved by using data recorders to record semi-structured interviews. External reliability, or the replicability of the study, means that the study's conclusions should depend on the subjects (or teachers) and the conditions (such as the school) rather than on the researcher. The research, therefore, needed to be documented to make clear how it was carried out and how conclusions were drawn from the data (Bakker & van Eerde, 2015).

The triangulation of several data sources was used to enhance the data's trustworthiness and credibility. The perceptions of the teachers implementing computing (semi-structured interviews), their teaching methods (observations) and management views on computing (semi-structured interviews with the headteacher and computing coordinator) were combined to enhance corroboration of emerging themes (Israel et al., 2015).

3.10 Ethical Considerations

Several ethical issues needed to be considered in relation to the current research. Ethical approval was obtained through the School of Education's Ethics committee in compliance with the University's Code of Research Conduct and Research Ethics process prior to conducting the research. Although the research was conducted within a primary school, it primarily focused on working with teachers and other adults within the school. As part of my regular academic role, I taught lessons at the school, and parents were aware that I was supporting teachers. Observations of teaching took place, and, by necessity, these involved observing how children responded to the teaching, but were designed not to affect children's learning.

As the proposed research took place within a one-form entry primary school, it required the use of a gatekeeper, in this instance, the head teacher, to facilitate access. As the research could have involved all of the teachers within the school, it was important that no teacher felt compelled to take part due to the status of the headteacher (Brooks et al., 2014). Individual teachers were free not to be involved in the study, and to state that they did not wish to be observed, and did not feel under pressure from peers (or those of a higher status) to consent or refuse to consent (Hammersley, 2017b).

Connections gained through working as a lecturer were utilised to gain access to the school in the study (Robson, 2011); the study also involved offering advice and professional development to the school. It could be argued that the offer of advice posed an ethical dilemma, as it influenced the research (Brooks et al., 2014). However, as this is design-based research with the

express aim of implementing a change, this concern is less significant. It was also important to ensure that the offer of CPD to the school did not mean that teachers felt compelled to take part in the research; the CPD was open to all teachers within the school, whether they agreed to be observed and interviewed or not (Brooks et al., 2014).

3.10.1 Informed consent and privacy

All study participants were fully informed about the nature of the study. All teachers of computing in the school were invited to participate in the study, but participation was not mandatory. Any teacher who wished not to be involved in the study, in whole or in part, could do so. Whilst the study took place in a primary school and involved observing teachers in classrooms, thereby examining children's learning, the children were not the explicit focus of the study or the observations. Parents were therefore informed of the research being carried out within the school and the presence of the researcher (see Appendix 5); however, they were not required to give consent (Brooks et al., 2014).

3.10.2 Confidentiality, Anonymity and Data Protection

In line with the General Data Protection Regulation (GDPR), which came into force in 2018, all participants were provided with a copy of the GDPR information sheet for participants (Appendix 5). As all teachers were identifiable to the researcher, the data collected cannot be considered anonymous; however, it was kept confidential. Nevertheless, while participants in the study were identifiable to the researcher, all efforts were made to ensure that participants could not be identified by anyone external to the study.

Confidentiality was ensured by not disclosing any information from any participant in a manner that would allow them to be identified or traced. When reporting qualitative data, the researcher ensured that although they knew which teachers had provided the information and could identify participants from the data, this connection was not publicly disclosed (Cohen et al., 2011). Pseudonyms were used when reporting comments made by different teachers in any publication, and a pseudonym also referred to the school. Ecological validity required that the characteristics of the one-form entry school be taken into account during this research, requiring a rich description of the school; an ethical tension could be seen through this, as the need to provide a rich description of the school for the research could lead to an increased likelihood that the school would be identifiable. Therefore, anonymity could be breached (Cohen et al., 2011). Extra care was taken to ensure that this did not happen. It was also possible that, due to the small size of the school, other participants might be able to identify comments made by other teachers; therefore, care was taken to ensure that no information was published that could negatively impact any participants.

All data collected was handled in a manner that ensured data protection. As emails were not considered secure, they were used only to contact the participants and explain the purpose of the study. All data collected was treated confidentially, and only for the original research (Flick, 2011). All interview recordings were accessible only to the interviewer and members of the research team and were stored on password-protected storage media. Transcripts were also stored securely. All data was only used for the study (Flick, 2011) and was stored on the university server.

3.10.3 Non-maleficence

Research should not harm the participants. Hammersley (2017b, p. 60) states that:

Questions asked in interviews or on questionnaires may be distressing, classroom observations may be intrusive, in the sense of causing problems for teachers or students, and reporting findings may have harmful consequences.

Care was therefore taken to ensure that teachers understood that observations were purely for the research and were not intended in any way to be related to their performance. Care was also taken to ensure that questions in interviews and questionnaires were not too personal or related to particularly sensitive topics (Swain, 2017).

3.11 Summary

This chapter has considered the rationale for conducting research into the picture of computing in a one-form entry primary school, with the intention of changing how computing is perceived and taught. The methodological approach, design-based research, has been discussed, along with the data corpus collected, including how it was gathered and analysed. Research validity, reliability and ethical considerations have been discussed. The following chapter presents the findings from both phases of the study and aims to interpret the gathered data.

CHAPTER 4: Data Analysis and Findings

The methodology chapter (Chapter 3) reviewed the rationale for conducting research and described the range of methods used; this chapter aims to describe and interpret the data gathered. Divided into two Phases, the first Phase aims to paint a rich picture of how computing was taught in the school at the start of the research period, exploring what is done well and any challenges to teaching computing and will aim to answer Research Question 1. The second Phase will address research questions 2-4 by exploring strategies to overcome challenges to teaching computing in a one-form entry primary school. Figure 4.1 illustrates the data collection process.

Figure 4.1

Data collected at The Watermill School from September 2017 to October 2019



4.1 Phase 1

This first phase of the findings describes what computing was like at The Watermill School at the beginning of the research period, reviewing what was being taught and the challenges the school faced related to the curriculum. A range of data was gathered to understand what computing entailed at the project's outset. Teachers completed written interviews to collect initial thoughts on computing. Following these, a learning walk was conducted with the computing coordinator to observe evidence of computing within the school. This involved visiting classrooms, searching for evidence of computing displays and observing the two teachers who had agreed to be observed. Follow-up semi-structured interviews were conducted with the head teacher, the computing coordinator, class teachers (n=4) and teaching assistants (n=2). Document analysis was conducted on the Watermill School's planned curriculum and termly curriculum overview documents, which were published on the school's website.

Exploring the data based on a range of themes, this section will aim to answer research question 1 (**RQ1**):

How is computing currently taught in the school? What are the key challenges?

4.2 What parts of the curriculum are being taught, and for how long?

At the Watermill School, there is a clear distinction between the National Curriculum, the school's planned curriculum, and the enacted curriculum.

4.2.1 The Planned Curriculum

The Watermill School's planned curriculum for 2017-2018 was compared with the 2014 English National Curriculum for computing (DfE, 2013c). Each National Curriculum (DofE, 2013) objective was given a code; these were then identified in the Watermill School's published planned Curriculum documents, with instances counted in NVivo.

Computational Thinking accounts for half of the national curriculum objectives, suggesting that it should be a significant proportion of computing teaching in schools. However, this has not been reflected in the school's 2017-18 computing curriculum, where Computational Thinking objectives comprise a third of the school's curriculum map and, therefore, a third of what teachers are expected to teach.

Table 4.1 shows the objectives in the curriculum map (25 in KS1 and 50 in KS2). This reveals that the computer science or Computational Thinking objectives have limited coverage. 'Algorithms', 'create and debug' and 'logical reasoning 1' in KS1 only appear 6 times, whilst in KS2, objectives, 'logical reasoning 2', 'programs' and 'variables' were found 14 times. The National Curriculum objective 'networks' did not appear in the school's planned curriculum.

Table 4.1

Instances of the National Curriculum Objectives in The Watermill School's planned curriculum.

National Curriculum strand	Key Stage 1 objectives	Objective occurrences in the school curriculum	Key Stage 2 objectives	Objective occurrences in the school curriculum
Computer science	Algorithms	1	Logical reasoning 2	7
	Create and debug	4	Programs	5
	Logical reasoning 1	1	Variables	2
Information Technology	Digital content	11	Networks	0
			Software	12
Digital Literacy (including e-safety).	Uses of technology	3	Search	13
	Safe technology	5	Acceptable behaviour	11

E-safety, comprising ‘acceptable behaviour’ and ‘safe technology’, appears 16 times and is strongly prioritised by the school. This is despite it only appearing once per key Stage in the National Curriculum. The school curriculum also states that the ‘Using Technology Strands are to be taught at every opportunity throughout the year with an intensive focus again in the summer term.’

4.2.2 The Enacted Curriculum

The school's block overviews⁹ and curriculum overview documents show a definite emphasis on e-safety, information technology and digital

⁹ The Watermill school divides their school year into four equal length blocks of teaching; the block overviews set out what objectives, for all subjects, will be taught during this time.

literacy skills in the school's enacted curriculum; for example, Chloe, teaching in Key Stage 2, mentions that:

We do e-safety every week, we do our sort of 20 minutes, and I try to also use them sometimes to cover the digital literacy things, so writing an email to Santa is, you know, sort of knowing how to send an email, so sort of cover both bases at the same time.

Figure 4.2

Block overview for Year 3 at The Watermill School

 <u>Maths and English</u> In Maths, over this first block we will be covering Place Value and Addition and Subtraction. We will also use our mathematical skills to interpret and present data based on our research of French speaking countries during French immersion week. In English, we will be retelling the well-known tale of Humpty Dumpty and learning to write letters based on work from the book 'Seal Surfer' by Michael Foreman. We will also be learning to write information texts as part of our study of fossils.	<u>SCIENCE - ROCKS AND SOILS</u> We will be discovering the fascinating secrets that rocks hold- how they are formed and how they shape our everyday lives. We will also be looking at the work of Mary Anning and becoming palaeontologists for the day! Additionally, we will learn how vital soil is to life on our planet. <u>HISTORY</u> We will learn all about life in the Stone Age, Bronze Age and Iron Age and the significant ways in which life changed for our hunter-gatherer ancestors. We will also be researching major sites of historic interest to find out just how archaeologists know so much about the past! <u>ART</u> We will create our own cave paintings, recreating techniques that would have been used in the Stone Age and focusing on mixing colours and using different types of marks. We will also undertake detailed observational drawings of fossils to further develop our sketching and drawing skills. <u>COMPUTING</u> We will be focusing on digital literacy skills, looking at the impact of our data online and how to keep it safe.	<u>PE - INVASION GAMES AND GYMNASTICS</u> We will be concentrating on developing our ball skills, specifically control, and also our communication skills in teams. We will also look at creating a range of balances, moving on to create sequences of movements. <u>FRENCH</u> We will be adding to our French vocabulary with colours, numbers and classroom instructions as well as learning some French songs and creating a map of Paris. <u>DT</u> We will be focusing on choosing the appropriate materials when designing for a purpose by making models of prehistoric shelters! <u>RE</u> In this block, our focus will be Christianity. We will be looking at the reasons people might choose to worship and what it means to them. 
---	--	---

The fact that e-safety is taught every week, even though it accounts for only a small part of the National Curriculum, shows the school's clear priority for ICT. Figure 4.2 (above) illustrates an example block overview for the first term of teaching in Year 3, with e-safety computing objectives identified as the sole objectives for the block. Computing was missing from the year 6 block

overviews for the first two academic year terms, indicating that it was not being planned for.

Whilst computing should be taught in each of the school's blocks, and therefore throughout the school year, this was not evident from lesson observations. During an early observation of children in LKS2 (age 7-8) using ScratchJr (ScratchJr, n.d.), an App aimed at 5-7-year-olds, on the iPads in the summer term, it was evident that this was only the second Computational Thinking lesson they had participated in that year. The process of handing out the iPads was also not a regular occurrence in the class, and it subsequently took up a considerable amount of time at the start of the lesson.

All teachers reported that computing was being taught for an hour a week. However, this did not align with what was observed either during formal teaching observations or when working to support teachers in the school. In the first instance, when I was scheduled to observe teachers (as part of a joint observation with Alex), an external visitor's assembly ran 20 minutes over, resulting in significantly shorter lessons than planned. This was a regular occurrence, with assemblies, visitors and other activities taking priority over computing lessons.

Throughout the KS2 lessons observed, children were taken out for small-group or individual work (interventions) focused on English or mathematics, meaning they missed parts of the computing sessions and had to 'catch up' when they returned. For example, during an afternoon lesson planned for computing with Year 6, over 30 minutes were spent editing English work and revising for an upcoming mental arithmetic test.

Hannah, an experienced teacher who was confident teaching ICT, recognised how English and mathematics can take precedence in a school's priorities when she mentioned during her interview that:

Well obviously English and maths takes priority and I think every other subject suffers from that, computing's one of the ones the children really enjoy and it's one of the ones where they actually feel really quite confident about but you have to make the effort to do it don't you but I don't think it gets the same amount of time or you know if something's got to slip it might well be computing to be honest...

Her observation that, despite children enjoying and feeling confident about computing, it is often not taught highlights how easily the subject can be relegated in favour of other subjects perceived as more important or easier to teach.

This analysis, therefore, reveals less computing than is required in schools' planned curricula and even less happening in the enacted curricula.

4.2.3 Who is teaching computing?

Although all teachers officially had responsibility for teaching computing to their classes, this was not what was observed in practice.

Alex frequently taught computing during teachers' PPA¹⁰ Time during the 2018-19 academic year. All teachers used this as an opportunity for him to

¹⁰ All teachers in English schools, including those in academies, are entitled to 10% of their timetable (usually an afternoon) allocated to planning and preparation time (PPA) where they are not in front of the class. Newly qualified teachers (NQTs) such as Matt receive 20%.

teach the Computational Thinking sections of the computing curriculum.

Hannah, a very experienced teacher who was computer co-coordinator in a previous school, mentioned this when discussing confidence in computing:

I handed over the stuff with the LEGO, all the algorithms and that kind of thing; Alex did that for me in year 4 this time.

This shows that even teachers who are confident in teaching the ICT parts of the curriculum will, given the opportunity, delegate the teaching of Computational Thinking to someone else.

On a couple of occasions, TAs were teaching parts of the curriculum; for example, while observing James teach UKS2, the TA was working with a group planning an e-safety drama, while he taught using LEGO WeDo (LEGO Education, n.d.). Teaching assistants were involved in a significant number of observed lessons; this was noted to be helpful during UKS2 lessons using the micro:bits (where they acted as an additional pair of hands to help with technical issues). However, sometimes they lacked the necessary skills to support children.

4.2.4 When is computing taught?

All observed lessons and self-reports from the teachers indicated that computing was being taught in the afternoon. The exceptions to this were instances where a more cross-curricular approach was employed, such as using the internet to gather information for English writing. This was always linked to the ICT parts of the curriculum rather than Computational Thinking.

4.2.5 How is computing taught?

Little teaching of Computational Thinking happened at the beginning of the research period, resulting in limited data collection. The few observed lessons demonstrated limited instruction of key concepts and minimal practical activity. When practical equipment was used, for example, LEGO WeDo (LEGO Education, n.d.), children and adults focused on building models and identifying missing pieces rather than developing Computational Thinking.

ScratchJr (ScratchJr, n.d.) lessons on the iPads with LKS2 mainly involved children working on set activities with little room for creativity. Once the iPads were handed out (a lengthy process), the children were mainly left to their own devices and did not receive instruction in using the app or key concepts.

This was something recognised by Rose, an experienced teacher close to retirement, when she mentioned that:

To be honest, it's just trial and error, they just poke and press things, and they learn loads of different strategies, they have different coding skills anyway, don't they...

The use of the words 'poke' and 'press things' indicates her awareness of a lack of specific instruction in her teaching. Whilst some children could figure out how to operate the app, this was often through trial and error, meaning they could not explain their thought process. There was also a lack of consistency about how key concepts were taught, with some teachers focusing on vocabulary whilst others engaged the children in projects.

ICT and digital literacy were taught more effectively. However, there was still an overemphasis on children watching an input (such as a video or PowerPoint) followed by working on a set activity or worksheet. The computing curriculum was taught mainly as standalone lessons rather than integrated into the rest of the children's learning, despite the school's thematic approach to learning. Exceptions to this were instances where the internet was used to research a topic, or ICT was utilised to present work using software such as Excel or Word.

Some teachers accessed schemes of work, including multipage lesson plans and videos, which could be complicated for non-specialists. James described finding these challenging to use at first:

There'd be these daunting 4- or 5-page lesson plans, and you'd find yourself stopping and starting a video... there was no rhythm... and it's stilted...

This led to difficulties maintaining the children's interest and disrupted the flow of lessons. There was no consistency in whether teachers used schemes of work, primarily due to personal preference. Whilst what was meant to be taught was in the planned curriculum, how it was taught was not prescribed or monitored.

The lack of consistency over how computing was taught was mentioned by Matt, the NQT, as being a barrier when he mentioned:

... making sure that we teach it consistently throughout the year groups, so however I'm teaching it, they're taught in the same manner and vice versa... that can be quite a bit of a barrier...

His recognition that not having a consistent approach to how or when computing was taught made the subject more challenging to teach. This highlights the importance of recognising how children can easily become confused by approaches or forget key points if they are not reinforced, and this issue needs to be addressed further.

4.2.6 Summary

The findings so far have identified gaps between the school's planned curriculum and the National Curriculum. The school's enacted curriculum overemphasises information technology and digital literacy. Whilst some teachers engage in computer science teaching, there is little practical activity. When given the opportunity, teachers will delegate Computational Thinking, and in some classes, it is not taught at all. The following section will, therefore, explore reasons for this, focusing on the challenges the teachers perceive.

4.3 What factors are perceived to influence the teaching of computing within the school? (Challenges)

All teachers, including the head teacher and computing coordinator, described barriers to teaching computing. Many of these challenges were also observed during teaching observations. A deductive coding scheme, adapted from Sentance and Csizmadia (2017), was employed to identify challenges in the data, with inductive codes added when an existing code did not provide an adequate fit. The identified challenges can be categorised into three areas: resources, students, and teachers.

4.3.1 Challenges related to resources

The main challenges relating to resources can be seen in Table 4.2

below:

Table 4.2

Physical Computing Challenges in The Watermill School

Term	Definition	Watermill School Example
Computing resource management	Challenges caused by allocating computing equipment to children.	Handing out laptops or iPads to children at the start of a lesson.
Technical problems	Challenges caused by outdated equipment or infrastructure.	Laptops not working or not connecting to the network.
Lack of equipment	Challenges are caused by either not having enough equipment or the right equipment.	Not having enough equipment for all children to access them at once or in small enough groups.

Technical problems and resource management can be considered together, as they have similar issues and consequences. Over half of teachers reported that equipment issues were a regular barrier to teaching and could prevent a lesson from taking place. This came up in conversation with Chloe when discussing why she felt less confident teaching computing compared to other subjects:

Okay, so I'm going to be really typical and say one of the things is just technology not necessarily always working, so you know, like there's been several times where I've gone to do a lesson and then half the laptops don't work and things like that.

Chloe's nervousness about technology not working, combined with her limited technical skills to troubleshoot when things go wrong, can slow the

pace of lessons or prevent them from happening altogether. The word "typical" suggests that she recognises this might be a common problem and would be off-putting to many teachers.

This was also a clear point mentioned by Sandra when discussing reasons why staff may be reluctant to teach computing:

Just confidence, I think, absolutely petrified of breaking things (laughs) and it not working, and frightened of looking stupid in front of a class of children when they can't work it.

Using emotive words such as 'petrified' and 'frightened' highlights the strength of emotions and how easily teachers can be deterred from teaching computing. She went on to state:

Frightened that you're going to bother to get it all ready and then it's not going to work, and so you're going to have a class of children in front of you, and it's not working, so even if you're not embarrassed, it's really frustrating, you can't deliver the curriculum like you wanted to.

This shows that even when teachers were not concerned about their subject knowledge, technical issues were a real barrier to teaching and could significantly affect how lessons were taught.

Outdated equipment was often perceived as a barrier to teaching computing. Whilst the school had invested in a new set of micro:bits towards the beginning of the research period, these were new to all staff (including Alex) and did not integrate well with the school's existing iPads. Connecting the micro:bits to the iPads became a recurring problem during lessons, with

instruction disrupted while teachers attempted to transfer students' programming from the iPads to the micro:bits.

Technical issues also occurred when a set of laptops was borrowed from the university for a model lesson; integrating the loaned laptops into the school's network proved difficult. The school's infrastructure meant that it took Alex and me 30 minutes to connect the university laptops to the school network, which is not uncommon and took time from Alex's lunchtime. For most teachers, this would be seen as a barrier to teaching computing, as without that commitment, the lesson could not have run. Issues with connecting technology were to recur throughout the research period.

All teachers mentioned a lack of equipment, which they considered a barrier rather than an opportunity. Teachers were particularly concerned about not having enough equipment for all children to participate in or access at once, without working in large groups. Sandra, however, acknowledged that with the proper training and confidence, it was perfectly possible to teach the computing curriculum without access to large quantities of equipment:

So, as always with IT, one of the biggest barriers is money and resources, but I don't think that we can use that as an excuse; I have to say I think it's perfectly possible to deliver the curriculum without all of those resources with a bit of creativity.

The use of the term 'excuse' suggests that attitudes towards teaching computing may have as much impact on whether it is taught as the actual equipment, and that this needs to be addressed further.

4.3.2 Challenges related to students

A range of challenges linked to the introduction of the 2014 curriculum (DfE, 2013c) were identified in the literature (see Table 4.3).

Table 4.3

Student challenges identified in the literature

Challenges relating to students	Example from the literature
Students not engaged	Children are not being engaged in computing.
Students lack confidence	Children often lack the confidence to try.
Students' problem-solving	Children often lack the skills to solve or debug problems independently.
Students starting too late.	Children have missed prior objectives due to the new curriculum or because the objectives were not adequately taught.
Students' use of language ¹¹	Children did not have the language to explain what was happening.

However, whilst all teachers mentioned children's responses to computing, no evidence was found of children being disengaged. When teaching classes, observing teaching, and in teacher interviews, all the teachers and teaching assistants expressed how engaged children were with all aspects of computing, especially the Computational Thinking sessions involving practical equipment. Amy, one of the TAs, for example, mentioned:

I think they love it, they love, you know, especially if it's iPads, everyone wants to use the iPads.

¹¹ Challenges in bold were inductive from the data.

This indicates that despite most children having access to technology at home, there was still excitement about using equipment in lessons.

The children were generally confident about their computing skills and were happy to give it a try. When introducing children to the micro:bits, they showed great excitement as they used the apparatus. However, most children lacked the resilience to problem-solve or debug when things did not work the first time.

The school recurrently highlighted children's use of computing language during observations and interviews. Despite key language being identified in the curriculum, most children struggled to explain their work and were unable to use the terminology of Computational Thinking. Alex, the computing co-ordinator, further recognised this in a discussion following a learning walk and lesson observation when he noted:

The thing I picked up was the language, particularly the year 3 but some of the year 5s as well, and I was saying to them 'so what are you actually making' and none of them were saying an algorithm or coding...

This demonstrates Alex's awareness of how children in the school use language and his recognition of the importance of using language in context.

The exception to the lack of key vocabulary use was children in Year 2, taught by Alex, who had been regularly exposed to computing language. This can be seen in the following conversation between Alex and Thomas in Year 2:

Thomas: *We were using the Bee-Bots*

Alex: *The Bee-Bots, so we made a series of instructions; what was the fancy name we used for those instructions? Can you remember there was a technical term we had for instructions?*

Thomas: *An algorithm*

Alex: *Excellent, and if something went wrong in that, then what did we do?*

Thomas: *We debugged it*

When observing Alex teaching Year 5 (age 9-10), he was confident using the language in front of the children. It was introduced clearly, and they were encouraged to use it when discussing their work.

4.3.3 Challenges related to teachers

While the majority of challenges relating to teachers described by Sentence and Csizmadia (2017) were found in the data, the most prevalent were a lack of training and/or subject knowledge, time, and confidence (see Table 4.4 below).

Table 4.4

Challenges related to teachers at The Watermill School

Challenges relating to teachers	Definition
Teachers' lack of confidence	Teachers are nervous about teaching computing.
Teachers and digital literacy	General confidence in using computers
Teachers and subject knowledge	Lacking sufficient subject knowledge to teach topics confidently.
Teachers and a lack of training	Teachers are lacking opportunities for professional development in computing.
Teachers' lack of time	Teachers often lack sufficient time to research and teach computing topics.

The teachers and TAs in the school had varied reactions to teaching computing. Whilst they all recognised its importance as a subject, enthusiasm varied, which was often linked to confidence.

There was a distinct split in the parts of the curriculum they preferred teaching. Teachers who taught under the old curriculum, for example, 'Hannah' and 'Rose', preferred teaching ICT skills, such as PowerPoint and Word, which were heavily featured in the former national curriculum.

Some teachers avoided teaching computing whenever possible. Caroline, the class teacher at the time in the year 6 classroom, who was both deputy head and mathematics coordinator, for example, was keen to have me come and teach a computing lesson and to be involved in maths support for children in the school, but was less interested in the actual teaching of computing. She delegated responsibility for the lesson to Hannah, the teacher responsible for her PPA sessions. This was a recurring theme throughout the research: when given the opportunity, teachers often delegated responsibility for teaching computing to someone else.

Organising teaching the lesson was in itself challenging, as can be seen from this part of an email from Caroline regarding organising a model lesson with the year 6 (age 10-11) class:

¹² Inductive from the data.

I've got in my diary that you were going to come and do some computing with my class on Wednesday afternoon this week. However, I remember you mentioning our laptops/notebooks and unfortunately we don't have our school ones available at the moment, as they have been taken away to be reformatted etc.

So should we see if we can reschedule once we have these again?

The email highlights the challenges of planning visits and underscores the need for additional resources in the school. Most early visits to the school required several emails, illustrating the challenge of contacting busy teachers who do not have access to email during the school day. Both time pressures and technical issues posed challenges to teaching lessons; without persistence, the lesson would likely not have taken place. The laptops mentioned by Caroline in the email above never reappeared at the school, illustrating the difficulty of teaching computing sessions that rely on specific technology.

Matt and James, however, both stated that they were excited about the opportunities presented by the new curriculum and were positive about teaching computing. Despite this, both thought that their subject knowledge was not as strong in Computational Thinking and had given these objectives to Alex to teach during their PPA time.

The most significant contributors to teachers' lack of confidence and/or reluctance to teach computing were insufficient training and/or subject knowledge. Even those teachers who felt confident teaching computing wanted more time to learn new equipment. Hannah, for example, explained that not knowing the equipment or having the confidence to try them were the main

reasons why Alex had taught the Computational Thinking objectives to her class.

Those teachers who had received training, such as Chloe, who had recently completed a computing course at university, still felt less confident because they had not had sufficient time to consolidate it. All teachers mentioned that CPD at school would be beneficial.

All teachers interviewed discussed the difficulties of the technical language, which, as James explains, can make teaching harder or even prevent it from being taught:

Well it's so unfamiliar as well because of all the things that were new this is a whole, literally a whole new language to learn and you can understand why there might be other teachers who are a bit more wary because if you feel you don't understand something the you're not going to put the time or effort into learning it and teaching it so it's very easy for it to get dropped off all together.

His mention of both the time and effort shows how busy teachers could easily feel overwhelmed by having to learn new terminology. The language of the curriculum emerged as a recurring theme throughout the research project and was a concern for the school.

4.3.4 Summary

Whilst some areas of the computing curriculum are being taught well at The Watermill School, there are apparent gaps in teaching computer science. The main challenges identified were technical infrastructure issues, teacher

confidence due to subject-knowledge gaps, and concerns about unfamiliarity with the equipment or equipment malfunctions.

Whilst student engagement and confidence in their ability to succeed at the Watermill School were not problems, their ability to explain what they were doing was. Language was a recurring theme throughout the data for both children and teachers, and was seen as a priority by many in the school.

4.4 Discussion of RQ1

In analysing the data related to Research Question 1, three main challenges were identified:

- 1) Computer science was not being taught consistently at The Watermill School.
- 2) Teachers lacked subject knowledge, which was coupled with the fear of not knowing how to use the equipment or the equipment not working.
- 3) Technical challenges due to outdated infrastructure and/or equipment.

In addition to the three main challenges identified, teachers at the school also found the language of the computing curriculum challenging to teach.

Whilst some areas of the computing curriculum were being taught at The Watermill School, there were gaps in teaching Computational Thinking, specifically the computer science objectives outlined in the English National Curriculum (DfE, 2013c). There was also an overemphasis on digital literacy and information technology in the school's planned and especially the enacted curricula. Teachers in the school all recognised the importance of teaching computing, but as in earlier studies (Larke, 2019; Sentance & Csizmadia, 2017; The Royal Society, 2017) focussed on those areas where they were more confident teaching. As teachers sit between the official curriculum and their local context, their decisions impact the planned and enacted curricula as they use, adapt or omit parts (Tran & O'Connor, 2024).

Most teachers identified subject knowledge as a factor that impacts their confidence when teaching computing. However, it is too simplistic to suggest that improving subject knowledge will automatically increase the school's teaching amount. As Philip and Garcia (2013) argue, when discussing the introduction of educational technologies, solutions that propose to 'fix the teacher' by increasing content knowledge or familiarity with the technology are rarely successful. The curriculum must be seen as relevant by teachers and as valuable to children. Larke (2019) argues that if teachers, as gatekeepers, do not see the relevance of the teaching curriculum, it will not be taught. Therefore, to enhance the teaching of computing at the Watermill School, it must align with the school's needs and be perceived as relevant.

The literature identified children's lack of engagement with computing as a concern (Sentance & Csizmadia, 2017). However, the complete opposite was found at The Watermill School. During initial observations and teaching of children in Year 6 (age 10-11) in June 2018, they were very excited to participate in computing lessons. Still, they struggled to problem-solve and lacked the language to explain what they were doing. The concepts introduced were new, even though they should have been taught computing under the revised curriculum for four years. Observations of children in Year 3 (age 7-8), who had successfully learned computing in Year 2 when taught by Alex, showed large gaps in their computing knowledge due to a nine-month period when they received no computing instruction. This suggests that a consistent approach, where computing concepts are revisited frequently, would be beneficial.

Without significant money inputs, it is impossible to alter the infrastructure or change the equipment available for teachers to use when teaching computing. Like many small schools (The Royal Society, 2017) pressure for space means that The Watermill School does not have a computer suite and only has outdated laptops and iPads that need replacement. Any solution, therefore, needs to consider this and be implementable within the current school environment.

Teachers expressed concern about the language used in the computing curriculum, and children struggled to articulate their thoughts on computing concepts. However, ensuring that language is not taught in isolation is important. Moschkovich (2015) argues that (mathematical) language taught in isolation ‘can have dire consequences’ and that vocabulary needs to be linked to conceptual understanding. Further research in mathematics suggests that using vocabulary to communicate mathematically is crucial for developing mathematical proficiency (Riccomini et al., 2015). When presented with new language, children need to connect to existing words or concepts they are more familiar with (Ingram & Andrews, 2018).

There are, therefore, three key areas that need to be addressed to enhance computing at The Watermill School:

- 1) Rewrite the school’s curriculum to cover all areas of the National Curriculum for Computing more equally (DofE, 2013), ensuring it is adapted to the school's needs.

- 2) Help teachers develop the subject knowledge to confidently and effectively teach the computing curriculum within the school's equipment constraints.
- 3) Develop an understanding of the language of computing shared by staff and students, ensuring this is treated holistically to ensure conceptual understanding.

The following section will, therefore, explore how this can be done within the school context.

4.5 Phase 2

Analysis of Research Question 1 revealed that while all teachers at the Watermill School recognised the importance of teaching computing, they struggled to engage with some parts of the curriculum, specifically the Computational Thinking or coding parts, which required more effective instruction. The school's planned and enacted curriculum overemphasised e-safety, information technology, and digital literacy skills. Factors, including resource issues and teacher confidence, also led teachers to focus on teaching the parts of the curriculum they felt most confident in, including ICT. Change was therefore needed if this situation was to improve. This next section reviews the second intervention Phase of the research, initially from a theoretical perspective to answer the second research question:

What approaches can support the school in overcoming these challenges?

As noted by several authors (Grootenboer et al., 2020; Harris & Jones, 2019; Lipscombe et al., 2020) the most effective way to enact change in a school is to work with middle or subject leaders. Middle leaders act as a buffer between school leaders and teaching staff; their unique understanding of their subject and the school's context enables them to influence what happens in their classrooms and the school, leading to more sustainable change. Grootenboer et al (2020) argued for an action-oriented professional learning approach, where collaborative work focuses on specific school issues to effect change. As an outsider, I needed to collaborate with a middle leader within the school to most effectively bring about change, ensuring that CPD was sustainable and led to lasting impact. The following section will therefore describe how I identified a middle leader to work with as a change agent within the school.

4.5.1 Working with a Change Agent

As the computing coordinator, Alex is the driving force behind computing within the school. He is passionate about teaching it well. He has the strongest subject knowledge of the teachers in the school and is most likely to take risks when teaching it. Nguyen et al.'s (2020) review of teacher leadership concluded that teachers' content, procedural knowledge, and motivation have a positive impact on their leadership ability. Therefore, Alex was a logical choice as a change agent to work with in enacting lasting changes to how computing, specifically Computational Thinking (CT), was being taught in the school.

While Alex was a class teacher at the start of the research period, he started working in a one-year part-time role with a charity at the beginning of

the second year of data collection. This meant that he temporarily reduced his teaching at the Watermill School. Somewhat unusually for a small primary school, cover for Planning, Preparation and Assessment (PPA¹³) at The Watermill School was taught by qualified teachers, so Alex took on this role. This meant he was teaching Year 2 one day a week and the other classes in the school for half a day a week, in 10-week blocks. All Watermill School teachers had a choice concerning the subjects they wanted Alex to teach. It is notable that all the teachers, including those with higher confidence in teaching computing, such as Matt, the newly qualified teacher in Year 2 (aged 6-7), chose to delegate the Computational Thinking parts of the computing curriculum to Alex while he was teaching their classes. Whilst this led to an opportunity for us to focus on the curriculum across the school and for Alex to extend his knowledge of physical computing when teaching the older year groups, it also meant that the rest of the teachers did not have to teach computing. It was also only a temporary solution to improving the teaching of computing within the school, as after one year, Alex returned to full-time teaching of a class, and teachers needed to take responsibility for teaching their classes.

The introduction of the micro:bits, purchased by the school towards the beginning of the research project, introduced a new component to Alex's teaching as these were a form of physical computing that he had not previously used. Alex's confidence in computing led him to adapt his teaching to use new

¹³ All teachers in UK schools are entitled to 10% PPA time in their school day, and newly qualified teachers (NQTs) get 20%. This is often taught by Higher Level Teaching Assistants (HLTAs).

resources. As Alex would be teaching the CT sections of the computing curriculum throughout the school, including using the new micro:bits that had been purchased, it made sense to initially focus on enhancing Alex's physical computing subject and pedagogical knowledge when teaching the older age groups. Alex had previously mainly taught Key Stage 1 (ages 5-7), and whilst he had good subject knowledge and was confident teaching computing, physical computing with the micro:bits was new to him.

Therefore, it was necessary to enhance Alex's subject and pedagogical knowledge of micro:bits so that he could teach them effectively during the project.

4.5.2 Curriculum change

Teachers draw on various sources when designing the intended curriculum, including national curricula, government guidance or recommendations, school-specific curriculum plans, and a range of instructional materials (Tran & O'Connor, 2024). Therefore, all teachers can be seen as curriculum makers through their interpretations of the National Curriculum (DofE, 2013).

A document analysis of the school's planned curriculum, in response to RQ1, revealed an overemphasis on the Information Technology and Digital Literacy strands, with less coverage of the Computer Science strand than would be expected based on the National Curriculum. This meant that, in addition to enhancing the subject knowledge of the school's teachers, significant changes were needed to the planned and enacted computing curriculum at The Watermill School so that it was more aligned with the goals

and objectives of the National Curriculum (DofE, 2013). This presented two challenges: working with Alex to redesign the school's planned curriculum and collaborating with teachers to implement it in their classrooms. In design-based research, collaborative design, also known as co-design, is one strategy for leveraging teachers' expertise to design, implement, and test educational innovations, thereby expanding teachers' agency through reform efforts (Severance et al., 2016). As computing coordinator, Alex was responsible for the school's planned curriculum, interpreting the National Curriculum with insight into the school's specific requirements, including the needs of children and the resources available. Teachers were then responsible for interpreting this in their planning and teaching it in the classroom.

Voogt et al. (2015) consider collaborative curriculum redesign a form of professional development, where teachers work with external experts to revise and rewrite curriculum documents. Teachers become agents of change by creating 'curricular activities and materials to improve student learning' (p. 262). As an external visitor to the school familiar with the curriculum, I was able to offer insight into it and provide a rationale for change, with the collaborative process allowing for reflection. The redesigned curriculum needed to be based on up-to-date knowledge of good practice and considered helpful by the teachers at The Watermill School if it was to be effective and lead to a change in what was being taught (Voogt, Laferrière, et al., 2015).

It was important to ensure any new planned curriculum was not too prescriptive; Tran and O'Connor (2024) discuss how prescriptive lesson plans and resource banks provided to teachers in Australia led to complaints of added stress and the de-professionalisation of teachers. Whilst providing lesson

plans aimed at saving teachers' time, they were not explicitly written for teachers' contexts and did not consider the needs of individual children in specific classes. More experienced teachers were less likely to use prescriptive lesson plans, preferring agency to plan in a way that best suited the children they taught. A similar outcome was observed at The Watermill School; teachers with greater computing subject knowledge, such as Alex, were more likely to adapt their teaching to suit their classes, whereas others, like James, were more likely to rely on lesson plans. Less confident teachers, therefore, needed support in converting the planned curriculum into an enacted curriculum that worked for their classes.

4.5.3 Career Professional Development in Schools

One way to enhance the teaching of computing within the school was to support teachers through Continuing Professional Development (CPD) focused on computing. CPD needed to be specific to the CT parts of the curriculum while allowing teachers to make links between computing and other subjects. The school were keen to teach computing in a cross-curricular manner, as this was how they taught the rest of the curriculum. Yadav et al. (2014) introduced CT concepts through concrete examples from everyday life, linking terminology to personal experiences; this approach could help teachers make cross-curricular connections and integrate CT into a busy timetable. It was also important to link computing pedagogy to the CPD, as teachers had found this more useful (Nugent et al., 2022) and not to focus too much on technical content teaching.

Kennedy's (2005, 2014) framework for CPD argues that collaborative professional enquiry is the most effective approach to school development.

Kennedy's (2014) transformative approach, which involves identifying a problem and following up with an activity to address it, can be carried out either with or without an external agent serving as a change agent. McNeill et al. (2016) discussed how an outsider could help to widen a school's knowledge base by bringing in expertise; as a university lecturer working with the school, I could help the school to develop sustainable change by bringing in computing expertise and through the development of CPD that was, 'flexible enough to fit in with the demands of an individual school's life,' (McNeill et al., 2016, p. 430).

This collaborative approach enables a school-based approach to CPD, working within the context of The Watermill School and its approach to teaching computing. As the UK Government's 2022-23 survey of educational technology in schools found that 67% of schools surveyed did not have enough time for CPD (IFF Research, 2023) the CPD needed to be taught in school, either as after-school in-service training (INSET) sessions or by working with teachers in their classrooms during the school day. This meant that teachers should be able to transfer their learning to working directly with their pupils (Durff & Carter, 2019).

McNeill et al. (2016) also argued that successful CPD programmes will incorporate a variety of instructional approaches, including peer work and classroom observation. A blended approach to CPD, with some elements of transmission, including a whole school professional development session, followed by more collaborative, transformative approaches such as collaborative professional enquiry through peer work and observation, was an

appropriate approach to develop computing subject knowledge and confidence to teach at The Watermill School (Kennedy, 2014).

The literature suggested that a lesson study or model lesson approach would be an effective form of collaborative professional enquiry to develop Alex's subject knowledge. Lesson study as professional development would involve joint planning of lessons before they are taught, followed by observation and discussion (Kim, 2021). Whereas a model lesson approach, such as that used by Sims et al. (2023) would involve one teacher planning a lesson, which the other observes for professional development. Both models can lead to professional development for both parties, although the model lesson approach focuses more on the observer's CPD needs. A combination of approaches was planned for The Watermill School.

4.5.4 Summary

Working collaboratively with a change agent at The Watermill School proved to be the most effective way to achieve long-term change. As the computing coordinator and the teacher most interested in the subject, Alex was the logical choice to work with. As computing coordinator, he was responsible for writing the school's planned curriculum and implementing changes, while also ensuring it was taught effectively. Involving the rest of the teaching staff in the design process was important at an enacted level to ensure collective responsibility and greater transformative action and learning (Voogt et al., 2015). The reviewed literature argued for a blended approach to CPD, combining a transmissive inset session that linked CT concepts to concrete examples with a more personalised approach, including lesson observations and peer working.

4.6 Research Question 3

The results of the analysis of RQ2 suggested that the best way to address challenges to teaching computing at The Watermill School was to work with a change agent within the school, first by rewriting the school's planned curriculum and second by enhancing teachers' subject knowledge and, hopefully, confidence through continuing professional development (CPD). The next part of the thesis will review how this was designed specifically for The Watermill School to answer research question 3:

How should these approaches be designed to be effective in this school?

4.6.1 Rewriting the Curriculum

Phase 1 of the research showed that computing needed to be taught more consistently across the school; the analysis of the planned curriculum (section 4.2.1) revealed an overemphasis on the curriculum's e-safety and Information Communication Technology (ICT) objectives, this being even though they constitute only half of the National Curriculum (DofE, 2013) objectives. Analysis of the enacted curriculum (section 4.2.2) revealed that teachers were more likely to teach ICT and e-safety, with Computational Thinking often delegated to other teachers or not taught at all. There also needed to be a shared understanding of what Computational Thinking meant to the school.

As computing coordinator, Alex was responsible for the school's planned curriculum. Following a discussion of the coverage of objectives in the curriculum and observations of class teaching, he rewrote the curriculum,

incorporating instructional material and resources available from Computing At School (CAS). As an external agent with greater insight into the curriculum, suitable software and hardware, and relevant instructional materials, my role was to provide guidance, rationale, and support for the change (Voogt, Laferrière, et al., 2015). Whilst I could have written a new curriculum directly, Alex's involvement in writing it was important due to his lived insight into the school, and it was intended to lead to greater transformative action.

Alex was keen to emphasise the language of computing, so he identified key vocabulary for each year group and the specific software and hardware to use when teaching. The three teaching strands were also renamed computer science, information technology, and digital literacy in line with the national curriculum. These replace 'Giving instructions,' 'Using Technology,' and 'Using Technology Safely,' which were previously used.

The reviewed literature indicated that whilst unplugged (Bell et al., 2009) approaches to teaching computing are helpful as they do not require a wide range of resources and remove the need for a computer suite (Curzon et al., 2014) they were most effective when combined with hands-on practical activities (Bell & Lodi, 2019). This approach was therefore used when redesigning the curriculum, which was designed to make the most of the school's existing resources, alongside a small outlay on new micro:bits. Practical computing was written into the new planned curriculum, with resources allocated to each year group.

The new planned curriculum was analysed in NVivo using the same coding scheme as in Section 4.2.1. Table 4.5 presents the revised curriculum map's

objectives in comparison to the original objectives. While there are a similar number of objectives for KS1 (23 in the new curriculum compared to 25 in the old), there are markedly more KS2 objectives in the updated curriculum (65 compared to the previous 50).

Table 4.5

Instances of the National Curriculum Objectives in the Watermill School's updated planned curriculum compared to the original curriculum

National Curriculum strand	Key Stage 1 objectives	Objective occurrences in the original school curriculum	Objective occurrences in the revised school curriculum	Key Stage 2 objectives	Objective occurrences in the original school curriculum	Objective occurrences in the revised school curriculum
Computer science	Algorithms	1	2	Logical reasoning 2	7	6
	Create and debug	4	5	Programs	5	10
	Logical reasoning 1	1	2	Variables	2	4
Information Technology	Digital content		5	Networks	0	2
		11		Software	12	13
Digital Literacy	Uses of technology	3	6	Search	13	15
	Safe technology	5	3	Acceptable behaviour	11	15

Of the 23 objectives identified for KS1, nine are now allocated to computer science (compared to the previous five), with the number assigned to Information Technology decreasing by six. Whilst the number of objectives for the Digital Literacy strand is similar, more are now allocated to 'Using technology', with fewer allocated to 'Safe Technology' (or e-safety).

Computer science, therefore, has more coverage in the Watermill School's updated planned curriculum than previously.

When reviewing the KS2 curriculum, the digital literacy strand still accounts for nearly half of the identified objectives (30 out of 65), with an even split between 'search' and 'acceptable behaviour'. There were more learning areas for Acceptable behaviour, or e-safety, than previously. The curriculum area of computer science, however, now has six additional objectives than the previous curriculum (20 compared to 14). There are a similar number of objectives identified for Information Technology; the National curriculum objective 'Networks,' which was missing from the previous planned curriculum, now appears twice.

While Computational Thinking objectives account for less than half of the school's objectives for either key stage (as might be expected based on the National Curriculum documents), these curriculum objectives receive more attention than in the previous curriculum. The planned curriculum also suggests that one of the six strands should be taught in each of the school's six teaching blocks, implying a more balanced coverage. However, it is suggested that Digital Literacy should be taught throughout the year, with an intensive focus in the summer term. This is a change from the previous curriculum, with Information Technology taught throughout the year.

The revised planned curriculum should mean that skills introduced in Year 1 (age 5-6) are built upon as the children progress throughout the school, with physical computing being introduced through Bee-Bots in Year 1. Software and technology previously planned for use in older age groups (such

as ScratchJr (ScratchJr, n.d.), previously taught in Year 4, for students aged 8-9) have been moved to younger year groups. Lego WeDo (LEGO Education, n.d.), previously used with UKS2 (9-11-year-olds), is now used with LKS2 (ages 7-9), and the newly purchased micro:bits are to be introduced from Year 5 (ages 9-10). While the changes to the planned curriculum better align it with the National Curriculum, they still require consistent teaching to have a substantial impact. The planned curriculum provides guidance for teachers that should lead to adjustments to their block plans and to the implementation of the curriculum.

The digital literacy strand still significantly emphasises ‘e-safety’. However, this is an understandable priority for Alex and the school. The new planned curriculum also identified key vocabulary to be taught, which Alex considered necessary.

4.6.2 Language in the Revised Curriculum

Part of the changes to The Watermills School’s planned computing curriculum involved identifying specific computing vocabulary from the National Curriculum (DfE, 2013c) that should be introduced to each year group. The language associated with computing was identified as a concern by all the teachers, and Alex felt that identifying key vocabulary for each year group would lead to its more consistent use throughout the school. Table 4.6 presents the key vocabulary identified for each year group. This indicates that most of the vocabulary identified for Key Stage 1 (3 words from 5) and Key Stage 2 (14 words from 18) is related to the computer science strand. There is no indication in the planned curriculum documents of how the vocabulary will be taught.

Table 4.6*Vocabulary to be taught per year group*

National Curriculum Strand	Year Group	Key Vocabulary Identified¹⁴
Computer Science	1	Instructions*, Programme*
	2	Instructions, Predict*, Programme
	3	Algorithm*, Control*, Debug*, Evaluate*, Sequence*
	4	Abstract*, Algorithm, Control, Invent*, Logic*
	5	Abstract, Algorithm, Decompose*, Evaluate*, Invent, Output*, Variables*
	6	Algorithm, Debug, Decompose, Evaluate, Logic, Pattern*
Information Technology	1	
	2	
	3	Data/ Information*
	4	Data/ Information
	5	Data, Hardware*, Input*
	6	Information/ Data, Input
Digital Literacy	1	Predict*
	2	Digital Footprint*, Information/ Data*
	3	Data/ Information, Digital Footprint
	4	Data/ Information, Input*
	5	Data/ Information, World Wide Web*
	6	

Including the language in the curriculum does not automatically imply that it will be taught. Although the data limitations restrict our inferences, the revised planned curriculum is expected to have altered teachers' planning for 2018-2019 of the research project (2018-2019).

In summary, the key changes identified in the new planned curriculum are:

- Renaming the teaching strands in line with the National Curriculum.

¹⁴ Asterix vocabulary indicates the first time the vocabulary should be introduced.

- Greater emphasis on Computational Thinking with physical computing taught throughout all year groups.
- Key vocabulary is identified to be taught throughout the year, with progression in the language used.
- Software and hardware identified for each year group, with progression in challenge.

The introduction of more physical computing, alongside the increased focus on Computational Thinking, whilst positive, meant additional challenges for the staff, some of whom, as previously described, were not confident in teaching computing. Therefore, if computing teaching were to be fully integrated into the school, more work with Alex as a change agent was needed to ensure that the planned curriculum changes led to enacted changes.

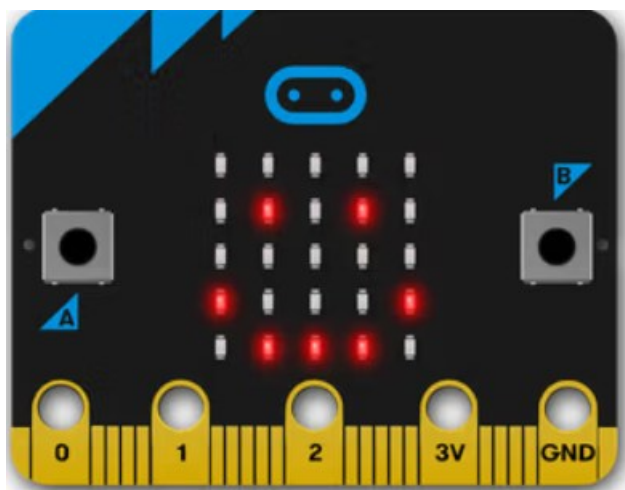
4.6.3 Teaching a Lesson

In the summer term of the first year of data collection (2017-2018, June 2018), I taught the outgoing Year 6 class (age 10-11) an introductory lesson to CT using a university set of laptops and micro:bits. The micro:bit, a pocket-sized programmable computer with LEDs (see Figure 4.3), can be programmed to display messages, feature programmable buttons, and a range of inputs. Initially introduced to secondary schools in 2016, it offers a practical approach for children to engage in physical computing (micro:bit.org, n.d.). At the time

(2018), micro:bits were routinely available in secondary schools (for students aged 11 and above), but were only starting to be used in primary schools.

Figure 4.3

The BBC micro:bit (micro:bit.org)



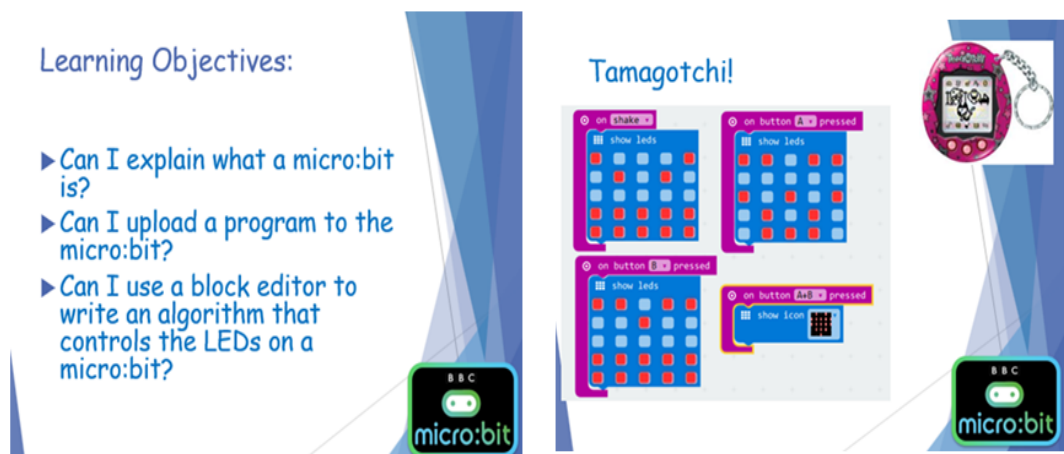
The lesson was taught in the summer term, after the children had completed their statutory testing in English and mathematics, and therefore had more time available for foundation subjects. The children would have been in Year 3 when the curriculum was introduced, so they should have been taught computing, and therefore CT, for nearly 4 years but had never seen micro:bits before and had limited physical computing experience; the limited amount of CT in the school's original planned curriculum meant that their previous practical computing experience was limited to using Bee-Bots in Year 2 (ages 6-7) and Lego WeDo in Year 5 (ages 9-10). However, some children may have engaged in informal learning experiences related to CT, such as participating in clubs or undertaking home learning.

The lesson included an initial introduction to the micro:bit and an opportunity for the children to write a basic programme on the micro:bit block

programming editor. The learning outcomes for the lesson, along with an example of basic programming using the text editor, are presented in Figure 4.4 below (a complete lesson plan is included in the appendix).

Figure 4.4

Illustrative slides from the taught lesson plan on An Introduction to micro:bits.



The children were very engaged in the session and showed genuine enthusiasm for computing; however, it was apparent that they needed a deeper understanding of the computer science concepts they were exploring. At the start of the session, only one child could explain what an algorithm was, and none knew what a program was. Teachers were pleased to see how well the children responded to the lesson; they listened carefully during the input, attempted to answer questions, and provided explanations for their reasoning.

Hannah, the PPA teacher present during the lesson, when asked about opportunities provided by teaching computing, during an interview, discussed the progress the children made:

I'm always impressed... with how quickly the children get used to whatever objective it is and whether I'm teaching it or not, for example, I was in the lesson that you taught year 6 when they were doing their coding, weren't they, and that was amazing, that progress was brilliant.

This demonstrates her recognition of how physical computing can engage children, as well as her awareness of the progress they have made. By the end of the session, the children could all use the block programming interface and write a basic program to upload to the micro:bits, although only some children were able to do this without assistance. Although not presented as CPD, the lesson provided insight into the challenges of teaching CT to children with little prior experience. It was valuable to me in assessing the children's needs and informing the content for subsequent CPD model lessons that I taught for Alex and James with pupils in Year 5.

The PPA teacher, Hannah, was present during the lesson, along with two teaching assistants who were typically in the class. Hannah, an experienced teacher, understood the ICT parts of the computing curriculum but admitted that she needed more subject knowledge of Computational Thinking. She fully engaged with the lesson, and at the end, she gave feedback that she had learnt a lot from participating. Additionally, as part of my job as a teacher educator, I met Hannah the following day and was informed that:

The children really loved that session you did with them yesterday... they've decided they want to use their fundraising money to buy a set (of micro:bits) for the school!

The use of fundraising money, traditionally allocated by the departing class to purchase a school gift, demonstrates the children's enthusiasm for the prospect of computing and suggests one way to engage children in Computational Thinking. Alex was also very interested in the concept of the micro:bits and included them in the rewritten planned curriculum for The Watermill School, to be used with Years 5 and 6 (Ages 9-11).

The successful teaching of the lesson, alongside comments from Hannah, suggested a good approach to developing CPD for Alex as the change agent.

4.6.4 Continuing Professional Development of The Change Agent

I designed a two-part approach to Alex's professional development. As previously described, curriculum co-design can be seen as a form of CPD; the initial process of rewriting the curriculum and sourcing better resources for teaching can therefore be seen as the first stage that helped to develop Alex's awareness of the curriculum, supporting his leadership through ownership of the new planned curriculum (Scornavacco et al., 2022). The second phase of professional development for Alex focused on his pedagogical and subject knowledge of teaching CT through programmable devices, specifically the micro:bits purchased in response to the lesson taught to Year 6, as described in Section 4.6.3.

The reviewed literature on lesson study and model lessons (such as Sims et al., 2023) argued that this approach can effectively enhance subject knowledge among primary school teachers. As previously discussed, Alex, identified as the logical change agent and collaborator, had good overall

computing subject knowledge and was confident teaching CT, but had not used the new micro:bits. This was therefore a logical focus for developing his subject knowledge of physical computing and his pedagogical awareness of how to teach Computational Thinking most effectively.

Alex and I were keen for the children's CT to be developed through a series of lessons, with the children growing and building on skills over time. We, therefore, decided to develop Alex's subject knowledge through a sequence of lessons, the first of which would involve me planning and teaching a model lesson to Year 5 (age 9-10) for Alex to observe, followed by lessons that we planned together and team-taught before Alex took over the planning and teaching. These sessions were intended to occur weekly while Alex was scheduled to teach the Year 5 class (ages 9-10) and focus on using the micro:bits.

The first lesson was adapted from the introductory lesson I had previously taught to Year 6 (see appendix). As the school had no working laptops, the lessons required the use of the school's class set of 15 iPads with the new micro:bits, so the children had to work in pairs. Using iPads rather than laptops meant using a previously installed BBC micro:bit APP for writing block programmes, which were then uploaded to the micro:bits via Bluetooth.

Before the first lesson, the children had not used the micro:bits and had limited experience with physical computing, beyond using Bee-Bots and Lego WeDo(LEGO Education, n.d.). While they had used ScratchJr (ScratchJr, n.d.) for a couple of lessons in Year 4, they had limited experience with block programming, which they would need to develop to fully utilise the micro:bits.

The fact that the revised curriculum was introduced while the children were in Year 4, without supplementary lessons to fill in knowledge gaps, contributed to the children's knowledge deficiencies. This needed to be incorporated into the lessons; the lesson started with questioning to elicit the children's understanding of key computational concepts before exploring a virtual micro:bit on the interactive whiteboard. The children then had the opportunity to explore the micro:bit app on the iPads before they used block programming to design a series of faces to display on the micro:bit in response to different inputs.

Alex fully involved himself in the lesson and worked alongside a group of children, discussing the best way to write their program and sharing their successes with them. Following the lesson, we discussed the next steps in the teaching process. The new planned curriculum for The Watermill School provides teachers with the objectives they need to teach. It suggests software and hardware but does not provide teachers with specific teaching plans or discuss the pedagogy to be used. Alex was keen for future lessons to be linked to the topic the children were learning that term, so we jointly planned a lesson to teach together, which involved the children designing a micro:bit interface to answer quiz questions about their history topic, making links and engaging them in both subjects. The final stage of the model lesson process involved Alex taking over the planning and teaching of cross-curricular micro:bit lessons for the class, which I then observed.

4.6.5 Initial Continuing Professional Development for the Staff

In Research Question One, I concluded that while teachers understood the importance of the computing curriculum, most teachers preferred to teach

ICT. Fear related to a lack of subject knowledge, alongside concerns about things not working, contributed to the Computational Thinking components of the curriculum either not being taught or being delegated to someone else.

I designed a two-part CPD process to further engage teachers in computing teaching by improving subject knowledge and increasing ownership of the curriculum. Following CPD for Alex, which focused on using the micro:bits and rewriting the curriculum to include more opportunities to teach Computational Thinking, an initial after-school professional development session was designed to inform the teachers of my research findings. Development time was then spent on updating knowledge of computing and pedagogy. This was planned to occur one month into the new academic year (2019-2020) when the 2014 curriculum had been mandatory for four years, although teachers had had access to it for five years. Arranging the session for October 2019 meant that teachers had had a month to get to know their new classes and would have the opportunity to apply their learning throughout the academic year. The session occurred during the teachers' usual weekly staff training/meeting time to ensure no extra requirement to attend beyond the mandatory after-school meetings. Whilst the session was open to TAs, none chose to participate, as this would have been outside their contracted working hours.

Six teachers attended the INSET¹⁵ session (n = 6), representing the six classes across KS1 (age 5-7) and KS2 (age 7-11); Caroline, who was teaching Reception at the time, did not attend the session, although she would typically have been present at all after-school sessions. Alex, at that point the Year 2 class teacher, helped facilitate the session by organising the participants and resources. The session took place in Alex's Year 2 classroom, utilising an interactive whiteboard. The participants sat in pairs to facilitate discussion. Two participants, Chloe (Year 3) and Matt (Year 4), were recently qualified and had only taught under the new curriculum. They had both received recent university training in computing, but their enthusiasm for teaching computing varied. The remaining participants (n = 4) had all been teaching for 5 years or more and had not attended recent continuing professional development (CPD) in computing. The exception was Alex, who, as the computing coordinator, had participated in a variety of computing education sessions.¹⁶

The initial CPD introductory session employed a fairly transmissive pedagogy. However, it was designed to encourage active participation by participants (see Appendix 4 for the full lesson plan used during the lesson). A PowerPoint presentation served as the focus of discussion. However, care was taken not to read directly from this, with regular activities built in to ensure engagement and demonstrate activities that could be used with children. The reviewed literature informed the design of the session (see Bean et al., 2015;

¹⁵ Inset or in-service education and training. In England, schools have 5 days where pupils do not attend, which are allocated to education and professional development. The term is also used to refer to teacher education sessions after school.

¹⁶ In the context, further information about the participants can be seen in Table 1.1.

Nugent et al., 2022; Yadav et al., 2014), ensuring a focus on pedagogical content knowledge while relating concepts to real-life and everyday contexts. Due to the school's focus on the language of computing, the first part of the session began with explaining the language alongside key concepts, ensuring that this was applied in a way that children would understand and linked to cross-curricular learning. For example, decomposition and pattern recognition concepts were linked to algebraic thinking and reasoning, as shown in Figure 4.5.

Figure 4.5

Slides used to demonstrate the concepts of decomposition and pattern recognition in mathematics.

Add up all the numbers from 1-200...

•30 seconds...

•starts...

Germa Rolfe email.g.rolfe@nmsu.ac.uk

Break the problem down into smaller pieces...

- What's $200 + 1..?$
- What's $199 + 2..?$
- What's $198 + 3..?$

Decomposition
Break a problem down into smaller pieces!

Pattern recognition
Finding similarities between things!

Germa Rolfe email.g.rolfe@nmsu.ac.uk

Teachers were also introduced to a range of unplugged (computer-free) activities they could use to teach CT and to ways to link them to key CT concepts.

The second stage of the CPD process focused on establishing connections between CT and the rest of the curriculum to help alleviate the time pressures identified in the English National Curriculum (DofE, 2013) in RQ1. The new planned curriculum provided teachers with the objectives they were required to teach, along with the software and hardware to use; however, it did not provide the context for teaching (which can change) or specify the pedagogy to be employed. Alex and Sandra, the Headteacher, wanted a cross-curricular approach to computing teaching that aligned with the teaching of the rest of the foundation and core subjects. Curriculum plans for the autumn term were then used, and teachers, working in Key Stage pairs, identified as many opportunities as possible to make realistic connections between the curriculum and their planning. An example of an unannotated curriculum plan is shown in Figure 4.6 below.

Figure 4.6

An example block curriculum plan from The Watermill School for Year 4



Maths and English

In Maths, over the first block we will be covering Place Value and Addition and Subtraction. There will be plenty of opportunities for children to be hands on with the resources to support and consolidate their understanding of number.

In English, we will have a spellings and phonics focus using the text *After The Fall*. This book is based on what happened to Humpty Dumpty after his famous fall! We will then move onto the text *Gorilla* by Anthony Browne where we will focus on writing in narrative style and from different viewpoints.

SCIENCE - FORCES

Our first topic will be about sound. We will explore how sound is made and complete investigations to observe how sound travels through vibrations.

HISTORY

We will learn all about the Ancient Greeks and their place in history. We will read myths and legends and learn about some of the Gods and Goddesses.

DT

We will research and study Greek pots before making our own clay pot. We will pay attention to finishing techniques that are in keeping with the Ancient Greeks.

COMPUTING

We will practise our programming skills as we use the program Scratch to make and test algorithms.

PE - SWIMMING

We will have weekly visits to the EPIC swimming pool (on a Thursday) where we will be continuing to develop swimming skills.

FRENCH

We will continue building our French knowledge and understanding through simple finger rhymes and listening to French stories.

MUSIC

In Block 1, we will be learning to play violins with our specialist music teacher.

RE

Islam will be our first RE focus this year. We will learn about the mosque and how it is used for worship.



Actively involving teachers in collaborative curriculum planning was designed to give teachers individual and joint ownership of the curriculum, leading to greater curriculum change (Voogt, Laferrière, et al., 2015) and increased professional development.

Staff were then offered the opportunity to participate in further continuing professional development (CPD), including additional support with curriculum planning and lesson observation. After the initial CPD session, three Key Stage 2 teachers requested additional support with lesson planning. Chloe, who had originally been the most anxious about teaching computing, was interested in support with a lesson linked to geography. Sally wanted to

integrate computing into her next history unit, and James was interested in help with planning and using micro:bits to teach Year 5.

4.6.6 Summary

Various approaches were designed to help The Watermill School overcome the perceived and identified challenges to teaching computing. The first focus was on Alex, as a change agent and computing coordinator, rewriting the curriculum to ensure balanced coverage of objectives, with CT objectives clearly identified. The school's focus on language was recognised, but care was taken to ensure this did not become an overarching priority. The new curriculum fed into termly planning. New equipment, a class set of micro:bits, was purchased for the updated planned curriculum.

Alex's subject knowledge of the new equipment was addressed next through a shared teaching approach; a series of lessons was planned where I planned and taught a CT micro:bit lesson to the Year 5 (age 9-10) class, followed by lessons that we planned together and team-taught before Alex took over the planning and teaching.

Finally, CPD was planned and implemented for the remaining teachers. It was informed by the reviewed literature and focused on improving teachers' subject knowledge by linking Computational Thinking and everyday experiences and integrating computing into the rest of the curriculum to alleviate time pressures and encourage a cross-curricular approach.

4.7 Research Question 4

In the following part of this thesis, I will explore the impact of changing the curriculum and implementing CPD, to see whether these led to

changes in how computing was taught in the school post-intervention to address the final research question (**RQ4**):

How do these designed solutions impact participants' experiences and outcomes?

4.7.1 The Curriculum

This next section will explore whether the rewritten planned curriculum at The Watermill School was being translated into teachers' planning. Individual teachers' interpretations of the school's planned curriculum are recorded in medium-term curriculum plans, written termly, that show an overall objective for the unit of work (see Figure 4.6 above). At The Watermill School, these are based around a theme or topic. Table 4.7, below, illustrates the termly main computing objective identified for each year group in Key Stage 1 (age 5-7) and Key Stage 2 (age 7-11) for the 2018-2019 academic year (2018-2019), after Alex had rewritten the curriculum but before CPD had been implemented for the rest of the school staff.

Table 4.7

*Computing curriculum objectives from teachers' medium-term plans.
Computer science objectives are shaded.*

	Term 1	Term 2	Term 3
Year 1	We are going to capture what we see on our 'explorer expeditions' by taking photos. We will take photos to match a caption as well as writing captions to match the photos. We will type sentences on a keyboard.	We will be using computers to control toys and talk about we can programme a screen turtle to move and change direction.	We will be developing our digital literacy skills by discussing the features of websites, including whether the information they contain is fiction or non-fiction.
Year 2	We are going to capture the design process of our planes in DT with photos and then add text and voice captions to explain each step.	In this block we will learn how to correctly use a keyboard and what to do if mistakes are made. Furthermore, within this unit we will look at editing pictures and consolidate our understanding of using sound applications to enable us to add them to our work.	In computing, we are going to be evaluate a set of instructions to see if they work.
Year 3	In computing we will be exploring how different devices are connected and how to use the term 'algorithm'.	During computing, we will be focusing on information technology, looking at how to take a photograph.	I can explain why people use a password and the benefits of sending an email.

Year 4	We will practise our programming skills as we use the program Scratch to make and test algorithms.	We will be learning how to create and 'debug' algorithms and also continuing to develop an understanding about e-safety and recognising what I need to do to keep myself safe online.	We will be learning how to decide if a statement on a website is true using our Roman topic and also how to select appropriate keywords to input in an internet search engine to get an accurate 'hit'.
Year 5	We will practice our programming skills as we use the program Scratch to make and test algorithms.	We will be working to develop our computer science skills using BBC Micros to develop different algorithms for different purposes.	In this unit, students will learn that systems are made of different parts that communicate with each other.
Year 6	Learning how to keep ourselves safe when using a phone and social media and to use a search engine effectively.	In computing, we will be developing our skills when using PowerPoint, learning how to use various animations and effects.	No computing on curriculum map

As described in the methodology, document analysis, using the same coding scheme used to analyse the planned curriculum at The Watermill School, shows greater coverage of computer science objectives, with at least one computer science (shaded) objective covered per year group in Years 1-5. Both Years 4 and 5 have two computing objectives.

In Key Stage 1 (ages 5-7), there was an emphasis on Computational Thinking taught through instructions, with Year 1 doing physical computing

using the Bee-Bots to map out the children's routes to school, which was linked to their topic 'Exploring Our School' to bring computing in line with the cross-curricular approach the school encouraged with other subjects. In Year 2, the emphasis was on evaluating written instructions for clarity, which was linked to English instruction writing.

In Key Stage 2 (ages 7-11), there was a greater emphasis on Computational Thinking, with the term 'algorithm' being explicitly mentioned. In Year 3, the term 'algorithm' was introduced, with children planning a sequence of algorithms to create programs linked to their rivers geography topic. Repeated loops were introduced in gymnastics, and instruction decomposition and debugging were brought into design technology. This continued in Year 4, where repeats and loops were reinforced through music lessons, and instructional algorithms and decomposition were linked to design technology. Scratch was used to teach computing in both Year 4 and Year 5, with a focus on designing and testing algorithms. The children in Year 5 were introduced to BBC micro:bits, building on the work started by Alex, and were also introduced to If-Then commands in obstacle courses during Physical Education (PE) lessons.

Year 6 had no computer science objective listed, and were missing computing in their curriculum plan for the summer term; this may have been due to statutory testing (SATs) in the summer term. Without observing every computing lesson, it is impossible to know whether the planned curriculum is being taught and fully implemented, or if the change will be sustained in the long term. However, increasing the focus on computing in teachers' medium-term planning is a positive step.

4.7.2 The Change Agent

The next section of this chapter considers the impact of working with Alex, as a change agent, on the teaching of computing in the school. Audio recordings of Alex teaching a 90-minute lesson to the Year 5 class using micro:bits, alongside field notes, were transcribed and analysed. As this study followed the same approach as Study 1 and used the same deductive coding scheme detailed in the methodology, which was developed from the literature (Sentance & Csizmadia, 2017), the addition of a new inductive code, ‘teachers and teaching approaches’, was made. This analysis reviewed the evidence for the original challenges to determine whether they were present in Alex’s teaching, either as challenges or as evidence of Alex’s successes following the intervention. Table 4.8 (below) lists the most frequently identified challenges in Alex’s teaching.

Table 4.8*Challenges related to teachers at The Watermill School, in order of occurrence*

Challenges relating to teachers	Definition`	Instances in the data
Teachers and teaching approaches ¹⁷	Teachers use different approaches to teaching computing	23
Teachers and subject knowledge	Teachers have the subject knowledge to teach topics confidently.	13
Teachers and troubleshooting	Teachers know what to do when things do not go to plan	7
Teachers' use of language	Teachers use the language associated with computing.	4
Teachers' lack of confidence	Teachers are nervous about teaching computing.	2
Teachers' lack of time	Teachers do not have enough time to research or teach computing topics.	1

Table 4.8 shows the changes in challenges related to teachers identified towards the end of the model lesson and team-teaching sequence of CPD with Alex. Whilst analysis conducted for Research Question 1 found that teachers'

¹⁷ New inductive challenges from the data.

confidence was the main challenge identified, this was not the case in Alex's teaching post-intervention. Although Alex was already the most confident teacher before intervention, he lacked the subject knowledge to teach using the micro:bits. The following analysis stage will explore the three main challenges: subject knowledge, teaching or pedagogical approaches, and troubleshooting, to determine the impact of working with Alex on his teaching of computing.

Subject Knowledge

Working with Alex on developing the computing curriculum at The Watermill School, through shared planning and team-teaching computing lessons using the micro:bits, seems to have positively impacted Alex's subject knowledge and general understanding of the curriculum. As discussed in section 4.3.2, Alex was already confident with the language associated with teaching Computational Thinking and had previously used it with the children in his classes; he built on this and used questioning to engage the children, as can be seen in this piece of teaching input where Alex was teaching the Year 5 (age 9-10) class:

Alex: Just remind me, who can remember what an algorithm was?

Let's give somebody different...

Benjamin: Instructions

Alex: Yes, that's the word, bang on there, then a set of instructions. An algorithm is a fancy term for instructions. And we're looking today at output.

We want our micro:bits to do something. We're going to put some instructions

in and it is going to output our desired response. What is our desired response, Alya? What do we want that micro:bit to do for us today?

Alya: We want it to say which answer we chose.

Alex: Excellent. So that's the output we want.

This illustrates how Alex used questioning to reinforce the children's understanding of key concepts rather than teaching the words in isolation.

Alex was able to use his increased subject knowledge to integrate the language of computing into his lessons, as can be seen in the extract from the lesson below:

And finally we're going to look at ... abstracting unnecessary commands. We're going to create commands. We're going to take them away and take away the unnecessary part.

This shows how Alex could link the key concepts and language to what was happening in the lesson, without explicitly focusing on the terminology.

Pedagogical approaches

Alex used various pedagogical approaches when teaching the children; he employed a lot of questioning and modelling, where he would demonstrate a skill before the children practised it. Whilst we had initially planned lessons using the micro:bit together and jointly taught them, Alex felt confident enough to prepare the lessons independently after the second lesson and used a similar pedagogical approach.

In the lesson extract below, Alex planned for the children to write a program to use the micro:bits to answer quiz questions linked to their history topic:

We're looking for a desired outcome and using our micro:bit to help you to answer. I'm going to put you in that quiz situation again and you're going to see if your desired output. The algorithm that you can create can support you with making that.

This illustrates Alex's adaptation of the school's planned curriculum to incorporate elements of both history and computing into a cross-curricular lesson.

Alex had the confidence to use mini plenaries with the children when he identified that they had not understood a concept, which can be seen in the extract below:

Team 5 can I just have your attention really quick, Five and four and Three, Two and one. I've just had a really good question. Can I put any number in, if in this quiz I'm only going to give you 3 answers. Is number 8 going to be useful?

This demonstrates that Alex had sufficient confidence in his subject knowledge to pause the children's work, teach a key skill that would advance their learning, and then allow them to continue.

Troubleshooting

Not knowing what to do if things went wrong, combined with a fear of 'looking silly', were identified in the literature and RQ1 as reasons teachers

did not engage with the Computational Thinking parts of the computing curriculum. As subject coordinator, Alex was already more confident using general computing technology and would be called upon to help if resources did not work. However, as previously discussed, physical computing with the micro:bits was a novel teaching approach. Following the model lesson and team teaching sequence, Alex was able to adapt his teaching when things did not go to plan, as can be seen in this section towards the end of the lesson, where the children were trying to download their programmes to the micro:bits:

Right, I've got it. Sorry guys. It's just some of them have just changed slightly when I practised with them. Which is fine.

Don't press anything; so what you need to do is when you get on to this screen, it's just press some of you is doing automatically. Some of you need to press Continue all your press is 'show how' put your hand up on myself or Miss Rolfe will come and help you.

They, just some of them just need to be re-synced I don't know why...

This shows how Alex's increased subject knowledge, combined with his confidence in teaching computing, enabled him to respond to challenges without disrupting the lesson. He was happy to admit to the children that he was learning too and explain his thinking to them, showing confidence in his knowledge and a positive approach to learning.

Alex identified that model lessons and team teaching had enhanced his subject knowledge and confidence. In an email sent to the rest of the teaching staff in September 2019 discussing extra support, he wrote:

... and we even team taught (sic) a bit last year, which was really useful

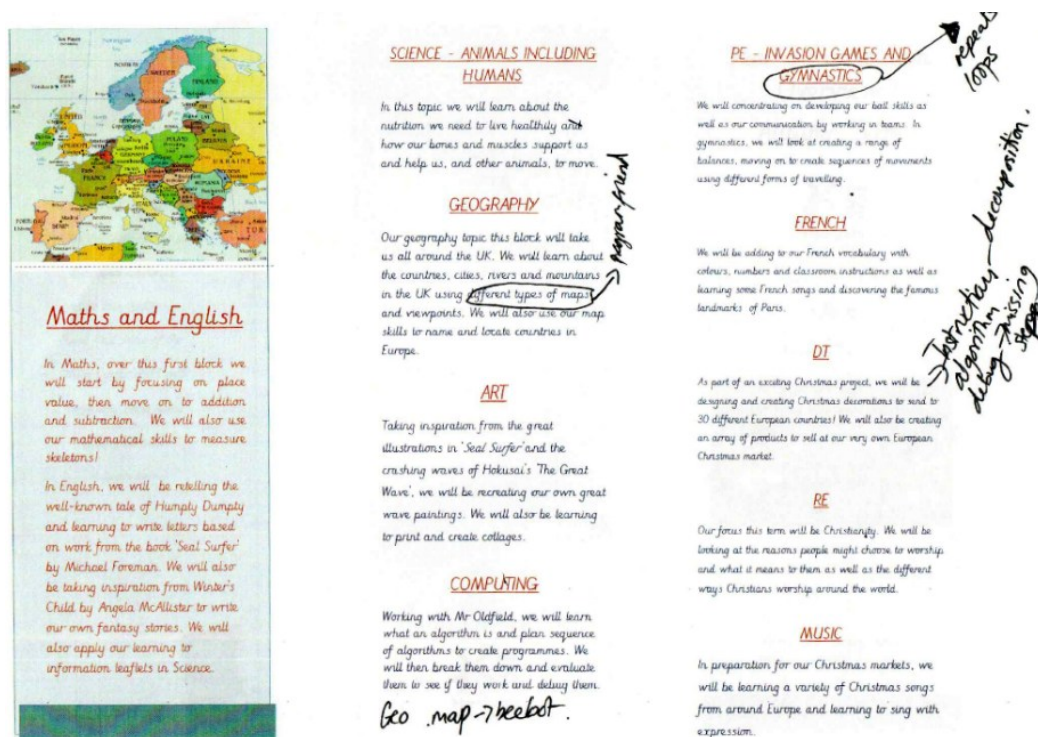
However, before the lesson, Alex had spent time experimenting with the micro:bits and was fully engaged as an active participant rather than a passive observer. It is possible that staff who were not as invested in teaching computing may engage in the process less actively and therefore gain less from it.

4.7.2 The Other Participants

Following the initial CPD for the school's teachers, there were initial positive signs of success. Document analysis of annotated curriculum plans, described in the methodology, revealed that all teachers could identify potential areas of the curriculum where they could incorporate CT objectives, such as linking loops to dance in PE in Year 1 and writing algorithms and debugging missing steps in Design Technology in Year 4. An example of an

Figure 4.7

An annotated curriculum plan for Year 3



annotated plan for the autumn term of Year 3 is shown in Figure 4.7 below.

This demonstrates how Matt identified that algorithms could be taught through unplugged instruction writing in geography, decomposition of instructions and debugging could be taught when writing instructions in design technology (DT), and repetition and loops could be taught during gymnastics.

Whilst James felt more comfortable about the concepts of CT, he was still uncomfortable about using the micro:bits independently and requested that I teach a lesson to show him and the children the basics of using the micro:bits. I, therefore, taught the refined introductory micro:bits lesson to James' Year 5 class (who had been in Year 3 at the start of the project and were the first class to be observed using iPads). The children demonstrated significantly more awareness of Computational Thinking than previous groups I had taught, which may be attributed to their consistent exposure to computing during the project.

As part of a routine email exchange with James, during which we discussed the lesson, he observed that he felt more confident about teaching the following steps:

Team 5 loved the lesson and can't wait for their next one. I learned loads, too, and feel much more confident teaching using the micro:bits.

This provides some evidence that model lessons can enhance the subject knowledge and confidence of a range of staff, not just specialists. Although it is clear that James successfully teaching with micro:bits would be more compelling.

4.7.4 Summary

The designed changes to the curriculum had an impact on how the computing objectives were translated into teachers' medium-term planning, with more Computational Thinking (CT) objectives evident across all year groups, except for Year 6.

Alex, the change agent, identified that shared lesson planning and team-teaching had a positive effect on his confidence when planning and teaching lessons using micro:bits. This was further evident in observations of lessons he taught, where he effectively integrated the micro:bits into cross-curricular lessons using a range of pedagogical approaches and was able to adjust his teaching and troubleshoot problems in response to technological issues.

The rest of the teaching staff initially responded positively to the CPD and were able to identify where CT could be integrated into their cross-curricular teaching. Additional model lessons with James also showed positive outcomes in his reported confidence.

My maternity leave, which coincided with the start of the COVID-19 pandemic, meant that I could not return to the school to conduct follow-up interviews with the teaching staff and carry out further analysis of the intervention's impact on computing teaching in the school. This will be discussed more in the next chapter.

CHAPTER 5: Discussion

5.1 Introduction

The research aimed to identify the challenges primary schools face in implementing the computing curriculum. It further aimed to develop and evaluate an intervention, through Continuing Professional Development (CPD), that would change how computing was perceived and taught in a specific one-form entry school. This concluding chapter summarises the extent to which this aim was met by discussing the key findings, considering how these findings relate to the literature, and the contribution made by the research. It will then discuss the study's limitations and provide recommendations for future research, before describing how the research could impact future practice.

5.2 What challenges do primary schools face in teaching the computing curriculum?

This first section of this chapter will discuss the picture of computing at The Watermill School at the beginning of the research. It will consider how the curriculum was planned and enacted in the school and what barriers there were to teaching it to answer research question one:

***RQ1** How is computing currently taught in the school? What are the key challenges?*

5.2.1 The Curriculum

The reviewed literature suggested that the transition from ICT to computing in the English curriculum, introduced alongside significant changes

to every subject at the primary level, presented challenges for teachers in redesigning curricula and acquiring new subject knowledge across all primary subjects. As previously discussed (Section 4.4), analysis of the data from The Watermill School suggested that this was also the case in this school, as they faced a range of challenges when implementing the revised 2014 computing curriculum. The empirical stage of the research began in 2017, when the new curriculum was expected to be in place for three years; however, there was little evidence of a change from the previous curriculum. There was an overemphasis on Information and Communication Technology (ICT) and E-Safety in the planned curriculum, with limited coverage of the Computational Thinking (CT) objectives. This was even more evident in the enacted curriculum, where computing was not consistently taught across the school. Whilst teachers at The Watermill School all recognised the importance of teaching computing, the majority of the teachers interviewed did not feel equipped to do so. This finding aligned with the wider teaching workforce at the time, as only 25% of teachers in the BT and Ipsos MORI (2016) survey respondents strongly agreed that they could prepare students for an increasingly digital world. The picture did not seem to have changed nearly 10 years after the introduction of the curriculum, as the UK Government's 2023 survey of teachers' working lives reported that only 39% of primary school teachers were confident about teaching computing (DfE, 2024c).

Some of the more experienced teachers perceived the computing curriculum's Computational Thinking (CT) objectives to be particularly challenging, as they mentioned in their interviews that the curriculum felt radically different from what had come before. Most teachers at The Watermill

School stated that they were much happier teaching the ICT and digital literacy objectives from the curriculum, especially since these were skills they were accustomed to using in their professional lives. This supports the broader literature of the time; for example, the Royal Society's (2017) large-scale survey of UK teachers found that many teachers reported being most confident in teaching the elements of the curriculum inherited from previous ICT courses. A similar outcome was found by Mannila et al. (2018) during the introduction of a new Swedish National Curriculum, where teachers had higher computer self-efficacy for more traditional ICT and digital literacy teaching, and were less confident teaching content creation and CT.

While a couple of teachers (including Alex, the change agent in the study) did teach CT frequently and with confidence, for the majority, discussions and observations of lessons showed it was not taught at all or had been left until the summer term, and was often taught through a process of trial and error, rather than with objectives being explicitly planned for. This finding is consistent with Larke's (2019) study, which also reported limited computing instruction during the research period; the little they observed was mainly taught by one teacher, who was the head of computing. However, when interviewing the teachers at The Watermill School, they did not have strong views about the inclusion of CT in the curriculum; this is unlike Larke's (2019) study, in which teachers chose not to teach a subject they felt was not developmentally appropriate.

Existing literature has advocated for practical activity when teaching CT (see for example, Hodges et al., 2020; Sullivan & Bers, 2016). The Watermill School had a small selection of practical computing resources,

including Bee-Bots and LEGO WeDo sets; however, observations of teaching and analysis of planning showed that these were rarely utilised. When practical activities were carried out (e.g. during an observation of Year 5 using LEGO WeDo, as reported in Section 4.2.4), the focus of the lesson was on building models and identifying missing pieces, rather than on Computational Thinking. Despite The Watermill School's cross-curricular approach to teaching, observations and analysis of planning showed that computing was often taught as a standalone subject, unless it was being used to support research in other subjects or to present work.

Sentence and Csizmadia's (2017) original study, conducted just before the introduction of the curriculum, which formed the framework and the deductive coding scheme for Phase 1 of the present study, divided challenges related to the introduction of the computing curriculum into those related to teachers, students, and resources. The next section of this chapter will therefore further consider the challenges reported by the teachers at The Watermill School in these three categories, relating them to other research.

5.2.2 Challenges relating to resources

Alongside the significant changes to the curriculum, the teachers at The Watermill School reported other factors that posed challenges to teaching computing, including technical difficulties and financial constraints.

Poor technological infrastructure in schools was initially cited as one of the most significant barriers to teaching computing in the original 2012 Royal Society Report, which preceded the introduction of the new curriculum (Furber, 2012). This was still the case at The Watermill School five years later.

Like many UK primary schools, space constraints at The Watermill School meant it had replaced its ICT suite with a set of laptops (which did not function during the study) and a set of outdated iPads, making it burdensome for teachers to download apps. TIMM's 2019 data, gathered at the same time as this study, indicated that only 24% of UK students had access to sufficient school computers they could use occasionally (Mullis et al., 2020). This does not appear to have improved, as the UK Government's 2022-23 survey of technology use in schools, conducted five years after empirical work began at The Watermill School, reported that around half of UK schools lacked sufficient laptops or tablets for children to use (IFF Research, 2023). It was also challenging to integrate outside equipment into the school's network, a problem common to most schools, which often have outdated networks alongside insufficient hardware and software for teaching computing (Sentance & Csizmadia, 2017).

There was discussion in my meetings with the computing coordinator about purchasing new devices partway through the research period, but this did not occur. Again, this is typical of the sector, as 95% of primary school head teachers questioned in 2022-2023 reported budgetary constraints as a barrier to increased technology use in schools (IFF Research, 2023). This situation is unlikely to change anytime soon. The rate at which technology, including educational technology, changes can be complex for schools to keep up with, both in terms of training to stay current with relevant software and Apps and the monetary costs of new equipment.

5.2.3 Challenges relating to students

In contrast to the literature (Sentance & Csizmadia, 2017), challenges at The Watermill School relating to the computing curriculum did not appear to be due to a lack of engagement or confidence among the children. The children at The Watermill School responded positively to computing lessons, a point noted by all the teachers and teaching assistants interviewed. Children in Years 5 and 6, in particular, responded enthusiastically to practical computing lessons during observations and appeared engaged in their learning (see Section 4.6.3). However, many children lacked the vocabulary to articulate their reasoning during observations, and others struggled with debugging when projects did not work as expected. These findings were consistent with those of Whyte (2021) who found that limited exposure to computing led learners to struggle to automate programmes and write concise algorithms in Scratch.

The challenges the children faced when writing programmes at The Watermill School may be linked to inconsistent teaching, as there were significant gaps in their computing education. While Sentance and Csizmadia (2017) explain this as being caused by students missing prior objectives due to their age at the point the new curriculum was introduced, this should not have been the case when the data were collected at The Watermill School, and is more likely due to a lack of consistency in teaching. Children in Year 3, observed at the beginning of the research period, for example (see Section 4.2.2), had been consistently taught computing by Alex (the change agent) in the previous year, but then had received little or no computing teaching until towards the end of the summer term in Year 3. This lack of consistency means

children may have to relearn key concepts that could have been reinforced if taught more frequently.

5.2.4 Challenges relating to teachers

The change to the curriculum led to additional challenges for teachers at The Watermill School, the most significant of which were a lack of confidence due to inadequate subject knowledge, concerns about equipment not working, and the time-consuming nature of teaching computing. These findings were consistent with the literature (including Larke, 2019; Sentance & Csizmadia, 2017; Sentance & Waite, 2018).

Although confidence was not explicitly measured in this study, most teachers mentioned it when discussing reasons why computing (specifically the CT objectives) was taught less successfully or not at all. This seems to be a common theme across the literature, with even teachers who express excitement about computer education being apprehensive about teaching it (see for example Israel et al., 2015).

The teachers at The Watermill School all reported that the most significant contributor to their lack of confidence was a lack of subject knowledge and/or awareness of approaches to teaching and using practical computing apparatus. Again, this aligns with the broader literature, as when asked to explain low confidence in teaching computing, a typical response from teachers surveyed by The Royal Society (2017, p. 55) was that they did not have ‘sufficient theoretical and technical knowledge of computing that included aspects of programming and coding.’ This picture does not seem to have changed 10 years on, with many secondary teachers interviewed as part

of a large-scale report into the future of computing education in England expressing that they felt unprepared to teach the computing curriculum and commenting on the variable quality and access to CPD (Kemp et al., 2024).

Even though all of the teachers at The Watermill School were responsible for teaching computing to their classes, interviews and observations (see Section 4.2.3) showed that, when given the opportunity, all the teachers in the school delegated the teaching of computing to either the computing coordinator (the change agent in this research) or another teacher or teaching assistant responsible for their PPA cover. This was irrespective of how confident they reported feeling about teaching computing, and was especially likely to involve delegating the CT objectives. The fact that the change agent in this research primarily covered teachers' PPA sessions during the second year of the study meant that this was even more likely to occur, leading to both opportunities and further challenges.

Fear that technology would not work and/or that equipment would break was also a significant contributor to confidence for over half of the teachers in the school, even when staff were not concerned about their subject knowledge. Larke's (2019) study found a similar outcome, with teachers struggling due to limited technical support, devices malfunctioning, or not functioning at all.

Alongside subject knowledge and confidence, time was seen by the school staff as the most significant barrier to teaching computing, encompassing the time spent planning and setting up computing lessons, as well as integrating computing into an already overburdened school week.

Whilst all of the teachers at The Watermill School reported that computing was taught once a week, this was not what was observed. Computing was usually taught in the afternoon and often got missed due to external visitors, sports competitions, or revision for statutory tests. This is consistent with Larke's (2019) study, in which teachers expressed concern about students performing well in SATs, indicating that they lacked external motivation to teach subjects like computing that were not tested or a focus of Ofsted inspections.

The practical nature of computing lessons, alongside the technical challenges, also means that time needs to be considered when planning them. On numerous occasions when teaching computing, Alex, as computing co-ordinator, needed to spend time before lessons ensuring that apps were downloaded onto iPads or that laptops were connected to the school's network. Whilst this is often considered part of the role of the computing subject leader, it took time out of his day, which may be a barrier to other teachers who may not have been as enthusiastic to start with.

The fact that the English National Curriculum was initially launched without dedicated support for teachers, in terms of both curriculum planning and resource provision, led to significant challenges both in the Watermill School and elsewhere. Teachers in schools were not provided with clear guidance on translating the objectives into developmentally appropriate lessons and felt they had not received sufficient support (Larke, 2019; The Royal Society, 2017). This meant that many teachers were left searching online for pedagogical guidance that was often not supported by research (Sentance & Waite, 2018). While some teachers at the Watermill School used schemes of work for their teaching, they were not well-received and could be confusing.

Prior studies have noted that teachers are generally willing to learn about computing and integrate it into their teaching if they receive support from people they perceive as experts (for example Israel et al., 2015). All the teachers at The Watermill School, including those who felt more confident teaching computing, mentioned during interviews and discussions that CPD would be beneficial in increasing their confidence.

5.2.5 Language

In addition to the challenges reported by Sentence and Csizmadia (2017), all the teachers at The Watermill School noted that the language of the computing curriculum posed a challenge for teaching. Alex, the computing coordinator, was confident using computing vocabulary in context, as evident from observing his teaching and conversations with children; however, this was not the case with the rest of the staff. This subsequently meant that most children struggled to explain what they were doing and could not use CT terminology.

Language can be considered a barrier to learning computing (Parry & Makin-Smith, 2024) which needs to be explicitly planned for, indeed, the Initial Teacher Training and Early Career Framework (ITTECF) (DfE, 2025b), which outlines what UK preservice teachers and early career teachers need to know and how to apply it, states in Standard 3 that pupils' language skills can be supported by teaching new words, meaning that pupils should be explicitly taught unfamiliar vocabulary.

Pre-teaching language, following a semantic wave structure, can help to overcome barriers caused by technical language by simplifying technical terms and abstract concepts (Waite et al., 2019). However, for teachers with lower confidence in teaching computing, it is possible that teaching may focus on instrumental rather than conceptual understanding of the language. As the National Centre for Computing Education (n.d.) states, vocabulary used correctly can unlock understanding, but when used inconsistently, it can lead to misconceptions that interfere with pupils' learning. This can especially be the case with words that have specific meanings in computer science, but different meanings in everyday conversation (Diethelm et al., 2018). It is therefore important for language to be used in context, with pupils given more opportunities to practice talking about computer science and consistent language used across the school (Diethelm et al., 2018).

5.2.6 Summary

It was clear that the introduction of computing into the National Curriculum had presented challenges for The Watermill School, both in planning and implementing it. Most teachers were especially anxious about teaching the Computational Thinking (CT) parts of the curriculum due to a lack of subject knowledge, and these parts had become marginalised. Technology caused concerns, with teachers anxious about it breaking and the time it took to set up. The Watermill School presented as a typical primary school in these regards, consistent with the wider literature describing English Schools at this time. Children, however, were overall excited by the subject and its opportunities. These results motivated the next phase of the research – how to support the school to enhance children's learning of computing.

5.3. How can we support schools in overcoming these challenges?

Phase 2 of this study aimed to understand how curriculum redesign and Continuing Professional Development (CPD) on the teaching of computing in a primary school could enhance student learning. A combined approach was needed to support The Watermill School in overcoming the barriers to effectively teaching the computing curriculum. First, the curriculum needed to be redesigned to meet students' needs, and second, CPD needed to be planned and implemented to increase subject knowledge and, therefore, teachers' confidence.

Participants' experiences were gathered through observations of teaching, follow-up discussions, document analysis of readily available curriculum documents and emails. It was found that the most effective way to achieve both of these goals was to work with a change agent or co-designer within the school, specifically the school's computing subject leader, Alex.

5.3.1 Curriculum redesign

A key finding of this research has been the challenges that teachers at The Watermill School faced in translating the National Curriculum (DofE, 2013) computing objectives into a planned and then enacted curriculum for their school. Whilst Alex, the change agent in this study, was interested in computing and was passionate about it being taught well, like the majority of teachers in primary schools, he was not trained as a subject specialist. Following the learning walk and subsequent discussions on the curriculum, he revised the curriculum (see Section 4.6.1) to ensure a greater balance across

the three strands of the computing curriculum, as well as identifying key vocabulary. The redesigned curriculum specifically identified software and technology to be taught in each key stage, with a greater emphasis on Computational Thinking, and practical computing taught throughout all year groups.

The school already had a class set of Bee-Bots, an example of educational robotics, which the teachers had experience in using. Research (such as Angeli & Valanides, 2020; Sullivan & Bers, 2016) has suggested that robotics can be usefully applied in teaching CT concepts, such as decomposing a problem, to children in Reception and Key Stage 1. These were planned for use in Key Stage 1 at The Watermill School; however, as Chevalier et al. (2020) note, careful scaffolding by teachers is required to ensure that children are taught CT concepts, rather than just learning through trial-and-error approaches. The planned curriculum for The Watermill School introduced visual or block programming (through Scratch) in Year 3 (ages 7-8) and moved Lego WeDo (previously taught in Year 5, ages 9-10) earlier, to Year 4 (ages 8-9). Both approaches are popular in UK primary schools, although again, research has argued that careful planning is needed to ensure that CT concepts are being actively taught (see for example Allsop, 2019; Benton et al., 2016, 2017).

The most significant change to the planned curriculum was the introduction of BBC micro:bits to Years 5 and 6 (ages 9-11). This followed the purchase of a set for the school in response to a lesson taught to the Year 6 class (as described in Section 4.6.3). Whilst microcomputers, like the BBC micro:bit, were initially designed to be used in secondary schools (Sentance,

Waite, Hodges, et al., 2017) their open-ended appeal, meaning students can engage in meaningful projects and goals, can be positive from a student's perspective (Hodges et al., 2020). As these were new to all staff, their introduction required careful consideration when planning CPD for both the change-agent and the rest of the staff.

A notable aim of the current research was to integrate computing, including Computational Thinking, across the curriculum, thereby aligning it with how other subjects are taught at The Watermill School and alleviating some of the time pressures of an already busy school timetable. Prior studies have noted that using a second set of knowledge practices (for example, a different subject area) may help to overcome the lack of time to teach each subject separately and may also increase children's motivation through using knowledge that is seen as more relevant or fun (Curzon et al., 2024). During the staff INSET session (see Section 4.6.5), teachers worked in pairs to identify opportunities to integrate computing into the other curriculum areas. Results indicated that rewriting the long-term curriculum at The Watermill School had some positive impact on how computing was planned within the school; including the staff in the redesign of the curriculum was key to its success. Analysis of the school's medium-term planning (see Section 4.71) after the intervention revealed more Computational Thinking concepts identified, which were linked to an overall topic.

Curriculum redesign, therefore, appears to have had a small positive impact on the integration of the computing curriculum into the rest of the school's planning and has increased the number of CT objectives in medium-term planning. However, introducing new concepts without support for

teachers would not be successful without addressing the other concerns, including subject knowledge and confidence, identified in response to RQ1. The following section will summarise additional work, with the change agent, that was required for a successful curriculum change.

5.3.2 Model lessons as a form of professional development

As previously discussed, rewriting the planned curriculum for The Watermill School had some positive impact on how computing was taught in the school. Introducing BBC micro:bits to the curriculum, however, had an impact on both the subject and the pedagogical knowledge required of teachers who would use them. During the second year of the study (2018-2019 2018-2019), Alex was mainly teaching PPA across the school (see Section 4.5.1) and therefore was teaching computing to the Year 5 class, which led to opportunities to develop his subject knowledge through a model lesson approach (as described in Section 4.6.4).

Model lessons were pedagogically effective when working with Alex to develop his subject knowledge, and he was able to successfully plan and teach using the micro:bits (see Section 4.7.2) following them. However, as a long-term solution, they would not be practical due to my involvement as a researcher being required. They also affected only one teacher's subject knowledge at a time, making them time-consuming to develop and deliver. In total, three teachers were involved in observing model lessons; however, it is uncertain whether these had a full impact on their long-term teaching, which will be further discussed in Section 5.3.3.

There are several possible ways that model lessons could be developed into a more long-term solution. Recent research, five years after empirical work at The Watermill School by Sims et al. (2023) utilised video model lessons to develop questioning with pre-service teachers, which could be a valid approach. However, as this took place in a classroom simulator, it would need to be carefully tested in a school setting, and the ethics of recording children would need to be considered. An alternative to model lessons being presented by an outsider would be to involve the change-agent in model lessons within the school, or to consider a lesson-study approach. These approaches will be discussed further in Section 5.6, which includes limitations and recommendations for further study.

5.3.2 Working with a co-designer

This next section will examine the impact of working with the computing coordinator as a change agent on computing within the school. Results from interviews, discussions, and observations of Alex (the change agent) suggest that his experience of rewriting the curriculum and participating in model lesson study-focused CPD was largely positive. He additionally reported that team teaching was useful (see Section 4.7.1).

Working with Alex on curriculum reform had a significant impact on how computing was taught at The Watermill School. I was fortunate to find a co-participant who was interested in the computing curriculum and was willing to dedicate his own time to working on the project (also see Section 4.5.1). Using Sentance et al.'s (2017) approaches to teaching computing, Alex can be seen as a computing 'Inspirer.' He was willing to take risks when teaching computing and used open-ended activities. He was happy to share resources

with other teachers and linked his computing lessons to real-world contexts, thereby encouraging independent learning. Alex was interested in computing, and I would argue that he taught it effectively, based on my informal observations of numerous schools visited as part of my job as a teacher educator, as well as through informal discussions with students. However, it is often the case that one teacher in a school will be passionate about a subject, such as computing, and will teach it well; another teacher, however, may not be as interested and will focus on a different area, meaning that computing gets taught less well, or not at all. This is even though all subjects in the curriculum are compulsory and should be taught equally, ensuring that all students have similar opportunities. The fact that computing was not taught at The Watermill School in Year 6 until the summer term, after the SATs, shows how an overemphasis on assessed subjects can marginalise the curriculum, squeezing out foundation subjects.

As Severance et al (2016) discuss, teachers in schools often lack control over school reforms and limited discretion as professionals to shape their practice through collaboration with colleagues. The very focus of reflection can lead to a change in practice; for example, my presence in the school, alongside dedicated development time, allowed Alex to focus on the curriculum. Through using Grootenboer et al.'s (2020) action-oriented professional learning approach to educational change, collaborative work, focused on local issues to develop slow systematic improvement, allowed us to work together on curriculum redesign and explicitly examine practice within the school, with the shared goal of changing how the computing curriculum was perceived and taught at The Watermill School.

The school's strong leadership (as acknowledged by Ofsted (Ofsted, 2018) led by Sandra, the headteacher, and Alex, as a middle leader, had a considerable impact on how computing was taught in the school. Strong teacher leadership makes it more likely that teachers will engage in workplace learning and share practice (Gümüş et al., 2022). Research into technology use when teaching (for example Durff & Carter, 2019; Ottenbreit-Leftwich et al., 2018), found that both peer discussion and strong mentorship had a positive impact on overcoming teachers' reluctance to use technology. As teachers who are reluctant to use technology may also be reluctant to teach computing, this is likely the case here as well.

However, the results also suggested that the Watermill School's redesigned curriculum still posed practical challenges due to ongoing equipment issues. While purchasing the BBC micro:bits and working through the curriculum had resolved some of the challenges described in Chapter 4, allocating resources remained time-consuming. The children were able to use the micro:bit interface to write their basic programs to control the virtual micro:bit display on the APP; however, even though Alex and I had both tested using the micro:bits with iPads, we experienced technical problems when trying to transfer the programs from the micro:bits to the school's older iPads. The limited number of iPads also meant that some children had to work in groups of three, sharing one iPad, which made discussion more challenging and had behavioural implications. Whilst Alex was able to overcome these issues (see Section 4.7.2), this may be more challenging for other staff. Results further indicated that professional challenges were also presented when teachers took on more responsibility for teaching computing, as some teachers

had not fully engaged in the CPD or still felt anxious about using the more advanced practical computing equipment.

This thesis has demonstrated that one effective teacher can have a significant impact on how a subject is taught in a school. Alex's ability to focus on teaching computing for a year led to a change. However, this is not a long-term solution, as few small primary schools can afford to hire subject specialists to teach dedicated subjects. The following section will therefore summarise the responses of the other teachers in the school to the intervention.

5.3.3 What were teachers' experiences of and outcomes from the intervention? CPD?

Before answering this question, it is essential to reflect on how the COVID-19 pandemic necessitated adjustments to the study's initial plan. I had intended to return to The Watermill School to work further with the teachers and conduct follow-up interviews after my maternity leave in March 2020; however, this coincided with the onset of the COVID-19 lockdown. Further restrictions in schools during the 2020-2021 academic year meant that even when in-person learning was taking place, the school did not allow visitors, including researchers, onto its premises. Empirical work on the study was therefore halted. This also had an impact on my professional life, as there were also challenges to teacher education at the university level, where most teaching moved online, and face-to-face observations of pre-service teachers ceased (and were replaced by online meetings with teachers to discuss progress). Unfortunately, this has placed limitations on what can be claimed about teachers' outcomes (see also Section 5.6).

Initial results from discussions, emails, and meetings with staff indicated a positive response to the CPD, and teachers had planned to incorporate computing into their teaching in a more cross-curricular manner. It had been hoped that involving the rest of the teaching staff in the design process through CPD, which encouraged them to rewrite their medium-term plans incorporating Computational Thinking, would lead to meaningful goals and new forms of classroom activity (Severance et al., 2016). Analysis of medium-term plans (also discussed in Section 5.3.1) showed evidence of a change in planning. However, it was not possible to carry out further observations of teaching or scrutinise children's work to see if this had had the desired impact. Two of the teachers requested additional support in adapting their computing lesson plans to be more cross-curricular (see Section 4.6.5), and one teacher asked to be involved in a model lesson. This suggests that, while the initial INSET session raised interest in teaching computing in a more cross-curricular way, further support would be needed to develop this approach in the school.

The fact that one of the teachers, Caroline, chose not to attend the CPD is interesting. All teachers are expected to attend after-school meetings, and time is put aside for INSET. As a senior leader in the school, she would primarily be expected to be involved. The fact that the session focused on computing is pertinent here; it would be very unusual for a teacher not to engage in CPD that was focused on subjects considered core, such as mathematics or English. However, the results of RQ1 (see Section 4.3.3) suggest that Caroline, despite being an experienced teacher, was the teacher who had the least interest or confidence in teaching computing. She repeatedly

delegated the teaching of computing to someone else during her planning and preparation time. This was the case both when Alex was responsible for her teaching sessions and in the 2018-2019 year, when Hannah taught her class. As other research suggests (for example, de Vries et al., 2013), teachers vary in the extent to which they engage with CPD, which can be linked to their beliefs about the nature of learning. Caroline may have had less student-centred beliefs about learning (de Vries et al., 2013) or her lack of interest and low confidence may have influenced her willingness to attend.

5.3.4 Summary

My research at The Watermill School had some positive impact on how computing was taught in the school, but also highlighted further challenges. Working with a middle leader as a change agent in the school was a positive way to impact change, and can impact on how curriculum change is perceived and enacted within a school. Rewriting the curriculum and developing model lessons with the computing coordinator had a significant impact on computing within the school. It impacted Alex's understanding of the computing curriculum and his teaching of the subject. Involving the rest of the staff in the project was less successful, which will be discussed further in Section 5.6, "Limitations of the Study and Recommendations for Future Work."

5.4 Contributions of this thesis

This research has contributed to the understanding of how computing is taught in small primary schools in England, broadening the picture of the challenges that teachers face and exploring the impact of introducing a new curriculum on the teachers at the school. Whilst these contributions are from a

single school and are not intended to be generalisable, they do provide contextual insights with potential relevance to similar primary school settings. Computing in primary education is a growing area of interest. Since I began this research, more researchers have examined it (such as, Curzon et al., 2024; Leonard et al., 2023; Sentance, 2021; Whyte, 2021); however, it remains relatively underexplored.

A further contribution of this research is to the existing literature on continuing professional development and curriculum change in computing, particularly the use of model lessons and lesson study as forms of professional development. Whilst there is research into teaching model lessons in other areas of STEM (such as, Vermunt et al., 2023; Wang et al., 2024), there is less focused on computing. This research has therefore contributed to the broader literature by successfully applying this approach in an area to which it has not previously been applied.

While remaining tentative about the implications of this work, three core aspects of research in particular seem worthy of note: the role of middle leaders as agents of curriculum and pedagogical change; implications for teaching pre-service teachers in teacher education programmes; and insights into how future national curriculum changes in computing might be enacted in primary schools. These will be discussed further.

5.4.1 Middle Leaders as Change Agents in Primary Computing

A key contribution of this research is its examination of the role of middle leaders in bringing about curriculum and pedagogical change. Working closely with the computing coordinator revealed the extent to which middle

leaders act as mediators between the official curriculum and the realities of everyday classroom practice. This study, therefore, illustrates that:

- Middle leaders can play a central role in supporting teachers to interpret and enact Computational Thinking and the curriculum in ways suited to their school context.
- Collaborative processes, such as shared planning, model lessons and opportunities for joint reflection, help teachers develop confidence with unfamiliar content; and
- Curriculum change is more likely to be successful when it is driven by staff who understand a school's ethos, constraints and pedagogical priorities.

These findings suggest that middle leaders are well-placed to champion computing and guide schools through incremental change. One possibility highlighted by the study is that sustainable improvement in primary computing may depend less on externally developed schemes and more on the presence of a knowledgeable, committed subject leader who can support teachers over time.

5.4.2 Supporting Novice Teachers to Teach Primary Computing

A second contribution of the thesis concerns the preparation and early professional development of novice teachers, both during initial teacher education (ITE) and for Early Career Teachers (ECTs). Newly qualified and recently qualified teachers in this study were enthusiastic about computing but remained uncertain about their subject knowledge and their ability to teach

Computational Thinking. Their experiences highlight implications for both Initial Teacher Education (ITE) and support during the ECT phase:

- Beginning teachers benefit from explicit modelling of Computational Thinking pedagogy, not just exposure to relevant computing resources.
- Opportunities to co-plan and co-teach with an experienced colleague support the transition from university-based learning to classroom practice; and
- Without structured support, novice teachers may avoid teaching computing or defer responsibility to more confident colleagues, impacting their long-term confidence.

The study, therefore, suggests that ITE providers and schools may need to offer more sustained, practice-focused experiences in teaching computing, particularly given the non-specialist nature of primary teachers. Ensuring that novice teachers encounter computing as an integrated, teachable part of the curriculum, rather than a technical specialism, may be essential for building confidence and competence.

5.4.3 Implications for Future Curriculum Reform in Schools

Finally, this research contributes to the broader discussion of how changes to the national curriculum in computing are interpreted and enacted in primary schools. The findings underline the challenges faced by schools when statutory expectations outpace available resources, guidance or teacher expertise. In particular, the study shows that:

- Schools frequently rely on internal leadership capacity to interpret new curriculum content.
- Without clear pedagogical guidance, teachers may focus on those aspects of the curriculum they feel most comfortable with, resulting in uneven coverage; and
- Curriculum co-design, undertaken collaboratively by teachers and a middle leader, can support the development of a more coherent and usable planned school curriculum.

This suggests that future changes to the computing curriculum, particularly in light of the recent curriculum reform (DfE, 2025a) and the new curriculum scheduled for introduction from spring 2027, are likely to be most successful if accompanied by meaningful opportunities for schools to engage in context-specific curriculum design. This would depend on teachers having both sufficient subject knowledge and dedicated time to lead curriculum development. While this study does not claim to present a universal model, it highlights that the successful enactment of curriculum reform depends on a balance of structured support and professional autonomy at the school level.

Taken together, these contributions highlight the importance of school-embedded approaches to curriculum development in computing. They underscore the pivotal role of middle leadership, the need for sustained support for novice teachers and the complex realities primary schools face when enacting curriculum change. While situated within a single school, these findings provide a basis for future work on how computing can be effectively and sustainably taught in primary schools.

5.5 Contributions to my professional practice

The most significant contribution of this research has been to my professional practice, both in my understanding of how children engage with computing and how teachers teach it. In chapter one, I positioned myself as a practitioner researcher, whose professional identity was shaped by over a decade of experience teaching in primary schools and later working in Initial Teacher Education. I described how observations from pre-service teachers returning from placement raised concerns about the limited teaching of computing in some primary schools, prompting my interest in understanding why this was happening and how it might be improved. Undertaking this research has not only deepened my understanding of these issues but also shaped my development as both a researcher and a professional. This section reflects on that journey.

5.5.1 Developing as a researcher

At the start of this professional doctorate, my approach to enquiry was closely linked to my identity as a teacher: practical, solution-focused and shaped by my experiences in the classroom and ITE. As I noted in Chapter One, my motivation for embarking on this research stemmed from a desire to ‘fix’ a problem I perceived in how the computing curriculum was being enacted in schools. Through the iterative, reflective, and theory-informed nature of design-based research, my perspective has shifted in several key ways.

First, I have developed a much stronger appreciation of the messiness of real-world educational settings and the need for methodological approaches

that recognise this complexity. Design-based research challenged me to move beyond simple problem-solving narratives and embrace an iterative way of working. I have learned that small, thoughtful shifts in practice, often arising from dialogue or reflection, can be as meaningful as more visible structural changes.

Secondly, over the course of my doctoral research, I have become more confident and intentional in my research approach. I have learned to work more closely with data, returning to it, questioning it, and noticing patterns that I might previously have overlooked. This has helped me to make decisions based on evidence rather than instinct alone.

Finally, I have gained an appreciation for the ethical and relational dimensions of research. Conducting research at a school where I was known professionally required care and ongoing role negotiations. I have become much more aware of my positionality as a researcher and the power dynamics of research relationships. I have therefore become more reflective about how my presence shapes the data and how to navigate that responsibility.

5.5.2 Developing as a practitioner

My identity as an educator has evolved throughout the research process. Regularly returning to a school environment has reminded me of the complexities and pressures involved in everyday teaching. In chapter one, I described a sense of frustration around computing not being taught. Working directly with teachers in schools has given me a greater appreciation of why this was the case. Understanding the fear, time pressures and competing

priorities of computing teaching has made me more empathetic and realistic in my expectations of both pre-service teachers and school partners.

This study has therefore directly impacted lecturers and seminars at the University where I work. I have developed a greater appreciation of the emotional aspects of teaching computing; teachers' apprehensions about 'breaking things' or 'looking silly' have led to sessions becoming more focused on building confidence, creating safe spaces for experimentation and positioning computing education as a developmental journey where practitioners will continue to engage with professional development throughout their careers.

In chapter one, I discussed my interest in teacher development and primary computing education, which was initially grounded in practice-based concerns. Through this study, I have developed greater confidence in contributing to curriculum conversations and advocating for a teacher-centred approach to professional development. Working alongside a computing coordinator has reinforced the significance of collaborative, situated change. Professionally, this has strengthened my conviction that sustainable improvement in computing capacity requires leadership, not merely resources or training materials.

5.5.3 Summary

In summary, this research has significantly shaped my professional practice by deepening my understanding of how children experience computing and how teachers manage its teaching in primary schools. Beginning the doctorate as a practitioner researcher, I was motivated by

concerns about the limited visibility of computing in some classrooms. Through design-based research, I developed a better appreciation of the complexity of school settings, grew more confident in working closely with data and became more aware of my positionality and the relational nature of research. Working alongside teachers has also reshaped my identity as a practitioner, helping me understand the emotional demands, practical pressures, and competing priorities that influence computing teaching. These insights have directly influenced my work in Initial Teacher Education, leading to a more empathetic, confidence building approaches and strengthening my commitment to collaborative, teacher-centred professional development. Overall, this study has developed me both as a researcher and as an educator.

5.6 Limitations and Recommendations for Further Research

This study has several limitations that need to be considered, along with associated recommendations for improvements or future work. This research was based on a single, one-form entry school, meaning that the results are specific to the local context of the school. The findings discuss the challenges in teaching computing reported by one set of teachers in a single school and describe an attempt to address these challenges in a specific context. The context of design-based research is crucial for the validity of the research. For ecological validity, a rich description of the context was provided. The unique nature of a single school, however, means that the results are not generalisable across multiple schools. However, this was arguably not the point of the thesis. Ideally, I would like to have expanded the research to investigate challenges in multiple schools, and future work would be needed to

explore how computing continuing professional development could be designed for different contexts.

The original research design involved returning to the school to examine the longer-term impact of the CPD on computing teaching within the school. The COVID-19 pandemic, which caused ‘unprecedented disruption to education systems around the world’ (Carroll & Constantinou, 2023), with its repeated restrictions on in-person learning, had an impact on the study's outcomes and the success of the intervention in changing the long-term teaching of computing within the school. As previously discussed in Section 5.3.3, I was unable to return to the school to conduct follow-up interviews or observe further teaching, which resulted in the completion of the empirical work earlier than I would have liked. While the move to online teaching forced teachers to utilise technology for teaching and learning, and could be seen as a catalyst for teacher innovation and development (Moorhouse & Wong, 2022), it is probable that some subject areas at The Watermill School were not taught as thoroughly as they should have been.

Whilst the majority of studies examining the impact of COVID-19 have focused on attainment in mathematics, reading, and writing (Education Endowment Foundation, 2022), Carroll and Constantinou’s (2023) large-scale survey of teachers’ experiences of teaching during the pandemic found changes in what was taught during remote teaching, with a focus on literacy and numeracy, and a reduction or cancellation of practical sessions in subjects such as science. This is likely to have been the case for computing at The Watermill School as well, and may also have impacted how computing was taught, as the focus may have shifted to teaching the ICT parts of the

curriculum, which could be used to present learning in other subjects. It is therefore difficult to determine the impact of CPD on the other teachers in the school, beyond what was initially observed. All teachers are likely to have had to use digital technologies to teach during the pandemic and have gained confidence in using them (Carroll and Constantinou, 2023). However, without ongoing practice of the concepts covered in the sessions, the gains will likely be lost.

There were also significant changes to the school staffing; despite the staff having remained stable (1 teacher retired and with an NQT appointed) during the three years of the project, the school's website showed that by September 2022, only 3 out of the original 7 class teachers Alex, Hannah and Sally, who was now deputy head, were still teaching in the school. Hannah and Sally were jointly teaching Year 6, meaning that five new teachers in the school had not been involved in the project or attended the CPD. This is not uncommon and should be considered for future work, as Clark-Wilson et al. (2019) discuss internal staff changes and teacher movement in and out of a school can have a long-term impact on the success of CPD.

Future work would be needed to develop the CPD approach into one that is sustainable within the school, without the presence of a researcher. While model lessons worked well with Alex and a confident computing co-ordinator could continue to work with teachers in the school to develop subject and pedagogical knowledge through model lessons, a lesson study approach (see, for example, Khan et al., 2020; McNeill et al., 2016; Vermunt et al., 2023), where teachers work in pairs or triads to develop their own practice, would have been more beneficial in encouraging the rest of the teachers at The

Watermill School to take more agency in their computing teaching. Schools often struggle to find time for CPD and have multiple priorities for development. Teacher collaboration and communities of practice may be a strong way to overcome this (Clarke & Pike, 2023; Parry & Makin-Smith, 2024) and would be a way to develop this research further and embed computing teaching in The School.

It seems clear that many teachers, not just at The Watermill School, find computing challenging to teach and lack the subject knowledge to do so effectively, which may in part be due to a lack of continuous professional development in schools (Parry & Makin-Smith, 2024). However, as Parry & Makin-Smith (20024) acknowledge, this should not be solely down to the school, and teachers must take responsibility for their own subject knowledge as part of recognising that teaching is part of a lifelong learning process. Work by professional organisations such as Computing At School (CAS) (2025) or The Raspberry Pi Foundation (2025a) including magazines and podcasts aimed at educators, such as Hello World (Raspberry Pi Foundation, 2025b) have helped fill computing subject knowledge gaps for primary education teachers. However, only those teachers who know where to look for support or are genuinely interested in the subject are likely to have taken advantage of these opportunities. Further work is therefore needed to address how teachers consider their own subject knowledge, not just in computing, and how they engage in continuous professional development throughout their careers.

5.7 Conclusions

I began this research with the goal of understanding why computing was not being effectively taught in all primary schools. The computing curriculum was introduced in 2014 with limited support for teachers at both the primary and secondary levels. As a result, teachers at The Watermill School struggled to adapt to the challenges of teaching a new subject that they did not fully understand themselves. As Parry & Makin-Smith (2024, p. 65) acknowledge, the words ‘Computational Thinking’, which are built into the National Curriculum, can lead to panic and teachers thinking, ‘I have no idea what that means, let alone how to teach it.’ This was further exacerbated at the school by a lack of funding to replace outdated equipment and the pressures of a full curriculum.

The designed intervention, through curriculum redesign and continuing professional development, aimed to address the challenges the school faced when introducing a new subject (computing) into the curriculum. By working with a change agent within the school, this research has hopefully made a small difference to how computing is taught in one school, but further work is needed to embed computing as a subject in all primary schools.

Computing in primary education in England is here to stay. Recent work suggests that the issues identified in this thesis, including teachers’ subject knowledge, technical problems, and a lack of funding, remain a concern for many schools. The work in this thesis may offer a partial solution to some of these.

REFERENCES

- Agyei, D. D., & Voogt, J. M. (2011). Exploring the potential of the will, skill, tool model in Ghana: Predicting prospective and practicing teachers' use of technology. *Computers & Education*, 56(1), 91-100.
<https://doi.org/10.1016/j.compedu.2010.08.017>
- Allsop, Y. (2019). Assessing computational thinking process using a multiple evaluation approach. *International Journal of Child-Computer Interaction*, 19, 30-55. <https://doi.org/https://doi.org/10.1016/j.ijcci.2018.10.004>
- Anderson, T., & Shattuck, J. (2012). Design-Based Research: A Decade of Progress in Education Research? *Educational Researcher*, 41(1), 16-25.
<https://doi.org/10.3102/0013189X11428813>
- Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior*, 105, 106185.
<https://doi.org/https://doi.org/10.1016/j.chb.2019.106185>
- Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, 105, 105954.
<https://doi.org/https://doi.org/10.1016/j.chb.2019.03.018>
- Bakker, A. (2018). *Design Research in Education: A Practical Guide for Early Career Researchers* (1st ed.). Routledge.
- Bakker, A., & van Eerde, H. A. A. (2015). An introduction to design-based research with an example from statistics education. In A. Bikner-Ahsbahr, C. Knipping, & N. Presmeg (Eds.), *Approaches to Qualitative Research in Mathematics Education: Examples of Methodology and Methods*. Springer.
- Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 84(2), 191-215.
- Barab, S. (2006). Design-Based Research: A Methodological Toolkit for the Learning Scientist. In R. K. Sawyer (Ed.), *The Cambridge Handbook of The Learning Sciences* (First ed., pp. 153-169). Cambridge University Press.
- Barab, S. (2014). Design-Based Research: A Methodological Toolkit for Engineering Change. In R. K. Sawyer (Ed.), *The Cambridge Handbook of The Learning Sciences* (Second ed., pp. 151-170). Cambridge University Press.
- Barab, S. (2022). Design-Based Research: A Methodological Toolkit for Engineering Change. In R. K. Sawyer (Ed.), *The Cambridge Handbook of the Learning Sciences* (3 ed., pp. 177-195). Cambridge University Press.
<https://doi.org/DOL:10.1017/9781108888295.012>
- Barab, S., & Squire, K. (2004). Design-Based Research: Putting a Stake in the Ground. *Journal of the Learning Sciences*, 13(1), 1-14.
https://doi.org/10.1207/s15327809jls1301_1
- Barefoot Computing. (2017). *Computational Thinking Poster*. Computing at School. Retrieved 05/02/2024 from
<https://www.barefootcomputing.org/resources/computational-thinking-poster>
- Barefoot Computing. (2021). *Computational Thinking Key Term Cards*. Retrieved 15/03/2024 from

<https://www.barefootcomputing.org/resources/computational-thinking-key-terms>

- Barni, D., Danioni, F., & Benevene, P. (2019). Teachers' Self-Efficacy: The Role of Personal Values and Motivations for Teaching [Brief Research Report]. *Frontiers in Psychology*, 10. <https://doi.org/10.3389/fpsyg.2019.01645>
- Barr, 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.
- BCS The Chartered Institute for IT. (2024). *Women are applying for Computing degrees in record numbers and slowly narrowing the diversity gap, new data shows*. BCS. Retrieved 28/05/2024 from
- Bean, N., Weese, J., Feldhausen, R., & Bell, R. S. (2015). Starting from scratch: Developing a pre-service teacher training program in computational thinking. 2015 IEEE Frontiers in Education Conference (FIE),
- Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer Science Unplugged: school students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13, 20-29.
- Bell, T., & Duncan, C. (2018). Teaching Computing in Primary Schools. In S. Sentance, E. Barendsen, & C. Schulte (Eds.), *Computer Science Education: Perspectives on Teaching and Learning in School* (pp. 131-150). Bloomsbury Academic.
- Bell, T., Duncan, C., & Atlas, J. (2016). *Teacher Feedback on Delivering Computational Thinking in Primary School* Proceedings of the 11th Workshop in Primary and Secondary Computing Education, Münster, Germany. <https://doi.org/10.1145/2978249.2978266>
- Bell, T., & Lodi, M. (2019). Constructing Computational Thinking Without Using Computers. *Constructivist Foundations*, 14, 342-351.
- Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2016). *Building mathematical knowledge with programming: insights from the ScratchMaths project*.
- Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging Primary Programming and Mathematics: Some Findings of Design Research in England. *Digital Experiences in Mathematics Education*, 3(2), 115-138. <https://doi.org/10.1007/s40751-017-0028-x>
- Berland, M., & Wilensky, U. (2015). Comparing Virtual and Physical Robotics Environments for Supporting Complex Systems and Computational Thinking. *Journal of Science Education and Technology*, 24(5), 628-647. <https://doi.org/10.1007/s10956-015-9552-x>
- Biesta, G. (2010). Pragmatism and the Philosophical Foundations of Mixed methods Research. In A. Tashakkori, C. B. Teddlie, A. Tashakkori, & C. Teddlie (Eds.), *SAGE Handbook of Mixed Methods in Social and Behavioral Research* (2nd Edition ed.). Thousand Oaks: SAGE Publications, Incorporated. <https://doi.org/10.4135/9781506335193>
- Blum, L., & Cortina, T. J. (2007). CS4HS: An Outreach program for high school CS teachers. Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education (SIGCSE'07),
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77-101. <https://doi.org/10.1191/1478088706qp0630a>
- Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 annual

- meeting of the American educational research association, Vancouver, Canada.
- Brinkmann, S., & Kvale, S. (2015). *Interviews: learning the craft of qualitative research interviewing* (Third ed.). SAGE.
- Brooks, R., te Riele, K., & Maguire, M. (2014). *Ethics and Education Research*. SAGE.
- BT and Ipsos MORI. (2016). *Tech literacy: A New Cornerstone of Modern Primary Education*. <https://www.bt.com/bt-plc/assets/documents/digital-impact-and-sustainability/skills-for-tomorrow/barefoot/ipsos-full.pdf>
- Bush, T. (2020). *Theories of Educational Leadership and Management* (Fifth Edition ed.). Sage.
- Busher, H., Harris, A., & Wise, C. (2000). *Subject leadership and school improvement*. Sage.
- Campanella, M., & Penuel, W. R. (2021). Design-Based Research in Educational Settings. In Z. A. Philippakos (Ed.), *Design-Based Research in Education*. Guilford Publications.
- Candappa, M. (2017). Case Studies. In *Designing Research in Education Concepts and Methodologies* (pp. 173-192). SAGE.
- Carroll, M., & Constantinou, F. (2023). *Teachers' experiences of teaching during the Covid-19 pandemic*. C. U. P. Assessment. <https://www.cambridgeassessment.org.uk/Images/682359-teachers-experiences-of-teaching-during-the-covid-19-pandemic.pdf>
- Catlin, D., & Woollard, J. (2014). *Educational Robots and Computational Thinking*.
- Chevalier, M., Giang, C., Piatti, A., & Mondada, F. (2020). Fostering computational thinking through educational robotics: a model for creative computational problem solving. *International Journal of STEM Education*, 7(1), 39. <https://doi.org/10.1186/s40594-020-00238-z>
- Ching, Y.-H., & Hsu, Y.-C. (2023). Educational Robotics for Developing Computational Thinking in Young Learners: A Systematic Review. *TechTrends*. <https://doi.org/10.1007/s11528-023-00841-1>
- Clark-Wilson, A., Noss, R., Hoyles, C., & Benton, L. (2019). Key Factors for Successfully Embedding a Programming Approach to the Primary Maths Curriculum at Scale. In J. Uffe Thomas, H.-P. Marja van den, & V. Michiel, [CERME11]. Eleventh Congress of the European Society for Research in Mathematics Education, Utrecht, Netherlands.
- Clarke, L., & Pike, S. (2023). The impact of forming a community of practice on teachers' attitudes and approaches to primary geography. *Irish educational studies*, 42(4), 911-930. <https://doi.org/10.1080/03323315.2023.2258493>
- Coe, R., Waring, M., Hedges, L. V., & Arthur, J. (2017). *Research Methods and Methodologies in Education SECOND EDITION*.
- Cohen, L., Manion, L., & Morrison, K. (2011). *Research Methods in Education*. Routledge.
- Cohen, L., Manion, L., & Morrison, K. (2018). *Research Methods in Education* (Eighth ed.). Routledge.
- Collins, A., Joseph, D., & Bielaczyc, K. (2004). Design Research: Theoretical and Methodological Issues. *Journal of the Learning Sciences*, 13(1), 15-42. https://doi.org/10.1207/s15327809jls1301_2
- Computing at School. (2025). *The community for computing teachers*.
- Computing at School Working Group. (2010). *Computing at school: The state of the nation A report of the Computing at School Working Group*. F. t. U. C. R. Committee. <http://www.ukcrc.org.uk/resources/briefings/computing.cfm?type=pdf>

- Creswell, J. W. (2014). *Research Design: qualitative, quantitative & mixed methods approaches* (4th ed.). Sage.
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Wollard, J. (2015). Computational Thinking: A Guide for Teachers. 1-18. Retrieved 21/03/2017, from <https://community.computingschool.org.uk/resources/2324>
- Curzon, P., McOwan, P., Plant, N., & Meagher, L. (2014). Introducing teachers to computational thinking using unplugged storytelling. WiPSCE,
- Curzon, P., Waite, J., & Maton, K. (2024). Teaching CS with and through other forms of knowledge. Proceedings of the 19th WiPSCE Conference on Primary and Secondary Computing Education Research, Munich, Germany.
- de Vries, S., van de Grift, W. J. C. M., & Jansen, E. P. W. A. (2013). Teachers' beliefs and continuing professional development. *Journal of educational administration*, 51(2), 213-231. <https://doi.org/10.1108/09578231311304715>
- Deng, Z. (2022). Powerful knowledge, educational potential and knowledge-rich curriculum: pushing the boundaries. *Journal of Curriculum Studies*, 54(5), 599-617. <https://doi.org/10.1080/00220272.2022.2089538>
- Denscombe, M. (2010). *Ground Rules for Social Research: Guidelines for Good Practice* Open University Press
- Department for Education and Employment. (1999). *Information and Communication Technology National Curriculum for England*. <https://webarchive.nationalarchives.gov.uk/ukgwa/20100209095326/http://curriculum.qcda.gov.uk/key-stages-1-and-2/subjects/ict/keystage2/index.aspx> Retrieved from <https://webarchive.nationalarchives.gov.uk/ukgwa/20100209095326/http://curriculum.qcda.gov.uk/key-stages-1-and-2/subjects/ict/keystage2/index.aspx>
- DfE. (2013a). *Consultation report: Changing ICT to Computing in the National Curriculum*. London: Department for Education Retrieved from <https://www.gov.uk/government/consultations/consultation-on-i-the-order-for-replacing-ict-with-computing-and-ii-the-regulations-for-disapplying-aspects-of-the-existing-national-curriculum>
- DfE. (2013b). *The national curriculum in England Key stages 1 and 2 framework document*. Department for Education
- DfE. (2013c). *National Curriculum in England: computing programmes of study*. (DFE-00171-2013). London: Department for Education Retrieved from <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>
- DfE. (2018). *School performance statistics*. gov.uk. Retrieved 14/12/2017 from <https://www.compare-school-performance.service.gov.uk/>
- DfE. (2023). *Changes to statutory induction for early career teachers (ECTs)*. <https://www.gov.uk/guidance/changes-to-statutory-induction-for-early-career-teachers-ects>
- DfE. (2024a). *Early years foundation stage statutory framework*. <https://www.gov.uk/government/publications/early-years-foundation-stage-framework--2>: Department for Education Retrieved from <https://www.gov.uk/government/publications/early-years-foundation-stage-framework--2>

- DfE. (2024b). *Schools, pupils and their characteristics Academic year 2023/24*.
<https://explore-education-statistics.service.gov.uk/find-statistics/school-pupils-and-their-characteristics>
- DfE. (2024c). *Working lives of teachers and leaders: wave 2 summary report*. Retrieved from <https://www.gov.uk/government/publications/working-lives-of-teachers-and-leaders-wave-2/working-lives-of-teachers-and-leaders-wave-2-summary-report>
- DfE. (2025a). *Curriculum and Assessment Review Final Report*.
<https://www.gov.uk/government/publications/curriculum-and-assessment-review-final-report>
- DfE. (2025b). *Initial teacher training and early career framework*.
<https://www.gov.uk/government/publications/initial-teacher-training-and-early-career-framework>: Department for Education
- Diethelm, I., Goschler, J., & Lampo, T. (2018). Language and Computing. In S. Sentance, E. Barendsen, & C. Schulte (Eds.), *Computer Science Education: Perspectives on teaching and learning in school*. Bloomsbury.
- DofE. (2013a). *Consultation report: Changing ICT to Computing in the National Curriculum*. London: Department for Education Retrieved from <https://www.gov.uk/government/consultations/consultation-on-i-the-order-for-replacing-ict-with-computing-and-ii-the-regulations-for-disapplying-aspects-of-the-existing-national-curriculum>
- DofE. (2013b). *National Curriculum in England: computing programmes of study*. (DFE-00171-2013). London: Department for Education Retrieved from <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>
- DofE. (2018). *Induction for Newly Qualified Teachers (England)*. DofE
- Donnelly, D., McGarr, O., & O'Reilly, J. (2011). A framework for teachers' integration of ICT into their classroom practice. *Computers & Education*, 57, 1469-1483.
- Durff, L., & Carter, M. (2019). Overcoming Second-Order Barriers to Technology Integration in K–5 Schools. *Journal of educational research and practice (Minneapolis, Minn.)*, 9(1), 246. <https://doi.org/10.5590/JERAP.2019.09.1.18>
- Dweck, C. S. (2017). *Mindset* (Updated edition. ed.). Robinson, an imprint of Constable & Robinson Ltd.
- Education Endowment Foundation. (2022). *The Impact of COVID-19 on Learning: A review of the evidence*. E. E. Foundation.
<https://educationendowmentfoundation.org.uk/guidance-for-teachers/covid-19-resources/best-evidence-on-impact-of-covid-19-on-pupil-attainment>
- Ertmer, P. A. (2005). Teacher Pedagogical Beliefs: The Final Frontier in Our Quest for Technology Integration? *Educational technology research and development*, 53(4), 25-39. <https://doi.org/10.1007/BF02504683>
- Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. On the iPad. *Journal of computer assisted learning*, 32(6), 576-593. <https://doi.org/10.1111/jcal.12155>
- Falloon, G. (2024). Advancing young students' computational thinking: An investigation of structured curriculum in early years primary schooling. *Computers & Education*, 216, 105045.
<https://doi.org/10.1016/j.compedu.2024.105045>
- Flick, U. (2011). *Introducing Research Methodology: A Beginner's Guide to Doing a Research Project*. SAGE.
- Flick, U., & Metzler, K. (2014). *An introduction to qualitative research* (5 ed.). SAGE.

- Furber, S. (2012). *Shut Down or Restart? The way forward for computing in UK schools*. T. R. Society.
<https://royalsociety.org/~media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf>
- Gomez, K., Kyza, E. A., & Mancevice, N. (2018). Participatory Design and the Learning Sciences. In F. Fischer (Ed.), *International handbook of the learning sciences*. Routledge.
- Gove, M. (2012). 'Harmful' ICT curriculum set to be dropped to make way for rigorous computer science <https://www.gov.uk/government/news/harmful-ict-curriculum-set-to-be-dropped-to-make-way-for-rigorous-computer-science>
- Grootenboer, P., Edwards-Groves, C., & Rönnerman, K. (2015). Leading practice development: voices from the middle. *Professional Development in Education*, 41(3), 508-526. <https://doi.org/10.1080/19415257.2014.924985>
- Grootenboer, P., Edwards-Groves, C., & Rönnerman, K. (2020). *Middle Leadership in Schools : A Practical Guide for Leading Learning*. Taylor & Francis Group.
<http://ebookcentral.proquest.com/lib/nottingham/detail.action?docID=6144274>
- Grover, S., & Pea, R. (2018). Computational Thinking: A Competency Whose Time Has Come. In S. Sentance, E. Barendsen, & C. Schulte (Eds.), *Computer Science Education: Perspectives on Teaching and Learning in School* (pp. 19-38). Bloomsbury Academic.
- Gümüş, S., Çağatay Kılınç, A., & Bellibaş, M. S. (2022). The relationship between teacher leadership capacity at school and teacher self-efficacy: the mediating role of teacher professional learning. *School Leadership & Management*, 42(5), 478-497. <https://doi.org/10.1080/13632434.2022.2123791>
- Guzdial, M. (2015). Learner-Centered Design of Computing Education: Research on Computing for Everyone. *Synthesis Lectures on Human-Centered Informatics*, 8, 1-165. <https://doi.org/10.2200/S00684ED1V01Y201511HCI033>
- Hammersley-Fletcher, L. (2002). Becoming a Subject Leader: What's In A Name? Subject Leadership in English Primary Schools. *School Leadership & Management*, 22(4), 407-420.
<https://doi.org/10.1080/1363243022000053420>
- Hammersley, M. (2017a). Research ethics. In R. Coe, M. Waring, L. V. Hedges, & J. Arthur (Eds.), *Research Methods & Methodologies in Education* (2nd Edition ed.). SAGE.
- Hammersley, M. (2017b). Research ethics. In R. Coe, M. Waring, L. V. Hedges, & J. Arthur (Eds.), *Research Methods & Methodologies in Education* (2nd ed., pp. 57-67). SAGE.
- Harris, A., & Jones, M. (2019). Teacher leadership and educational change. *School Leadership & Management*, 39(2), 123-126.
<https://doi.org/10.1080/13632434.2019.1574964>
- Hodges, S., Sentance, S., Finney, J., & Ball, T. (2020). Physical Computing: A Key Element of Modern Computer Science Education. *Computer*, 53(4), 20-30.
<https://doi.org/10.1109/MC.2019.2935058>
- IFF Research. (2023). *2022-23 Technology in Schools Survey*. Department for Education Retrieved from
<https://www.gov.uk/government/publications/technology-in-schools-survey-report-2022-to-2023>
- Ingram, J., & Andrews, N. (2018). Use and meaning: What students are doing with specialised vocabulary. IV ERME Topic Conference 'Classroom-based research on mathematics and language', Dresde, Germany.

- Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, 82, 263-279. <https://doi.org/10.1016/j.compedu.2014.11.022>
- Jackson, K., & Bazeley, P. (2019). *Qualitative Data Analysis with NVIVO* (3rd ed.). SAGE.
- Johnson, R. B., & Christensen, L. B. (2024). *Educational Research* (8th ed.). SAGE Publications. <https://bookshelf.vitalsource.com/books/9781071869949>
- Kalelioglu, F., & Sentance, S. (2020). Teaching with physical computing in school: the case of the micro:bit. *Education and Information Technologies*, 25(4), 2577-2603. <https://doi.org/10.1007/s10639-019-10080-8>
- Kemp, P., Wong, B., Hamer, J., & Copsey-Blake, M. (2024). *The future of computing education: Considerations for policy, curriculum and practice*. K. s. C. L. a. U. o. Reading. <https://scari.sites.er.kcl.ac.uk/#>
- Kennedy, A. (2005). Models of Continuing Professional Development: a framework for analysis. *Journal of In-Service Education*, 31(2), 235-250. <https://doi.org/10.1080/13674580500200277>
- Kennedy, A. (2014). Understanding continuing professional development: the need for theory to impact on policy and practice. *Professional Development in Education*, 40(5), 688-697. <https://doi.org/10.1080/19415257.2014.955122>
- Khan, I. A., Iftikhar, M., Hussain, S. S., Rehman, A., Gul, N., Jadoon, W., & Nazir, B. (2020). Redesign and validation of a computer programming course using Inductive Teaching Method. *PloS one*, 15(6), e0233716. <https://doi.org/10.1371/journal.pone.0233716>
- Kim, J. (2021). Through foreign eyes. A critical understanding of Lesson Study-based teacher education in Japan. In J. Kim, N. Yoshida, S. Iwata, & H. Kawaguchi (Eds.), *Lesson Study-Based Teacher Education*. Routledge.
- Kim, J., Yoshida, N., Iwata, S., & Kawaguchi, H. (2021). *Lesson study-based teacher education [electronic resource] : the potential of the Japanese approach in global settings / Edited by Jongsung Kim, Nariakira Yoshida, Shotaro Iwata, Hiromi Kawaguchi*. Routledge.
- Kite, V., Park, S., & Wiebe, E. (2021). The Code-Centric Nature of Computational Thinking Education: A Review of Trends and Issues in Computational Thinking Education Research. *SAGE Open*, 11(2), 21582440211016418. <https://doi.org/10.1177/21582440211016418>
- Kwon, K., Jeon, M., Guo, M., Yan, G., Kim, J., Ottenbreit-Leftwich, A. T., & Brush, T. A. (2023). Computational thinking practices: Lessons learned from a problem-based curriculum in primary education. *Journal of Research on Technology in Education*, 55(4), 590-607. <https://doi.org/10.1080/15391523.2021.2014372>
- Kwon, K., Ottenbreit-Leftwich, A. T., Brush, T. A., Jeon, M., & Yan, G. (2021). Integration of problem-based learning in elementary computer science education: effects on computational thinking and attitudes. *Educational technology research and development*, 69(5), 2761-2787. <https://doi.org/10.1007/s11423-021-10034-3>
- Kwon, K., Ottenbreit-Leftwich, A. T., Sari, A. R., Khlaif, Z., Zhu, M., Nadir, H., & Gok, F. (2019). Teachers' Self-efficacy Matters: Exploring the Integration of Mobile Computing Device in Middle Schools. *TechTrends*, 63(6), 682-692. <https://doi.org/10.1007/s11528-019-00402-5>
- Kyza, E. A., & Nicolaidou, I. (2017). Co-designing reform-based online inquiry learning environments as a situated approach to teachers' professional development. *CoDesign*, 13(4), 261-286. <https://doi.org/10.1080/15710882.2016.1209528>

- Larke, L. R. (2019). Agentic Neglect: Teachers as Gatekeepers of England's National Computing Curriculum. *British journal of educational technology*, 50(3), 1137-1150.
- LEGO Education. (n.d.). *LEGO® Education WeDo 2.0 Core Set*. LEGO Education. Retrieved 16/03/2026 from <https://education.lego.com/en-gb/products/lego-education-wedo-2-0-core-set/45300/>
- Leonard, H. C., Childs, K., Waite, J., Whyte, R., & Sentance, S. (2023). *Engaging primary (K-5) computing teachers in culturally relevant pedagogy through professional development* Proceedings of the 2023 Conference on United Kingdom & Ireland Computing Education Research, Swansea, Wales Uk. <https://doi.org/10.1145/3610969.3611118>
- Lipscombe, K., Tindall-Ford, S., & Grootenboer, P. (2020). Middle leading and influence in two Australian schools. *Educational management, administration & leadership*, 48(6), 1063-1079. <https://doi.org/10.1177/1741143219880324>
- Luo, F., Antonenko, P. D., & Davis, E. C. (2020). Exploring the evolution of two girls' conceptions and practices in computational thinking in science. *Computers and education*, 146, 103759. <https://doi.org/10.1016/j.compedu.2019.103759>
- Mannila, L., Nordén, L.-Å., & Pears, A. (2018). Digital Competence, Teacher Self-Efficacy and Training Needs. New York, NY, USA.
- McNeill, J., Butt, G., & Armstrong, A. (2016). Developing collaborative approaches to enhance the professional development of primary mathematics teachers. *Education 3-13*, 44(4), 426-441. <https://doi.org/10.1080/03004279.2014.973896>
- McNiff, J. (2017). *Action Research*. SAGE.
- micro:bit.org. (n.d). Retrieved 12.02.25 from <https://microbit.org/>
- Morreale, P., & Joiner, D. (2011). Reaching future computer scientists. In (Vol. 54, pp. 121-124): ACM.
- Moschkovich, J. N. (2015). Academic literacy in mathematics for English Learners. *The Journal of Mathematical Behavior*, 40, 43-62. <https://doi.org/https://doi.org/10.1016/j.jmathb.2015.01.005>
- Mullis, I. V. S., Martin, M. O., Foy, P., Kelly, D. L., & Fishbein, B. (2020). *TIMSS 2019 International Results in Mathematics and Science*. <https://timssandpirls.bc.edu/timss2019/international-results/>
- National Centre for Computing Education. (n.d.). *Primary Computing Glossary*. Retrieved 12/12/2024 from <https://teachcomputing.org/primary-computing-glossary>
- Nguyen, D., Harris, A., & Ng, D. (2020). A review of the empirical research on teacher leadership (2003–2017). *Journal of educational administration*, 58(1), 60-80. <https://doi.org/10.1108/JEA-02-2018-0023>
- Nugent, G., Chen, K., Soh, L.-K., Choi, D., Trainin, G., & Smith, W. (2022). *Developing K-8 Computer Science Teachers' Content Knowledge, Self-efficacy, and Attitudes through Evidence-based Professional Development*. New York, NY, USA.
- Ofsted. (2011). *ICT in schools:2008 to 2011*. <https://www.gov.uk/government/publications/ict-in-schools-2008-to-2011>
Retrieved from <https://www.gov.uk/government/publications/ict-in-schools-2008-to-2011>
- Ofsted. (2014). *School Inspection Report*. London: Ofsted
- Ofsted. (2018). *School Inspection Report*. London: Ofsted

- Ofsted. (2019). *Inspecting the curriculum*. (190024). Ofsted Retrieved from https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/814685/Inspecting_the_curriculum.pdf
- Ofsted. (2022). *Research review series: computing*. <https://www.gov.uk/government/publications/research-review-series-computing> Retrieved from <https://www.gov.uk/government/publications/research-review-series-computing>
- Ottenbreit-Leftwich, A., Liao, J. Y.-C., Sadik, O., & Ertmer, P. (2018). Evolution of Teachers' Technology Integration Knowledge, Beliefs, and Practices: How Can We Support Beginning Teachers Use of Technology? *Journal of Research on Technology in Education*, 50(4), 282-304. <https://doi.org/10.1080/15391523.2018.1487350>
- Palaiologou, I., & Kingdon, Z. (2022). What matters in early childhood? In C. Carden (Ed.), *Primary Teaching* (2nd ed.). Learning Matters.
- Parry, M., & Makin-Smith, K. (2024). Computing. In C. Mosey & J. Stothard (Eds.), *Planning for Adaptive Teaching in The Primary Curriculum* Learning Matters.
- Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (2020). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? *Computers in Human Behavior*, 105, 105849. <https://doi.org/https://doi.org/10.1016/j.chb.2018.12.027>
- Petko, D. (2012). Teachers' pedagogical beliefs and their use of digital media in classrooms: Sharpening the focus of the 'will, skill, tool' model and integrating teachers' constructivist orientations. *Computers & Education*, 58, 1351-1359.
- Peyton-Jones, S., Mitchell, B., & Humphreys, S. (2013). *Computing at school in the UK: from guerrilla to gorilla*.
- Philip, T., & Garcia, A. (2013). The Importance of Still Teaching the iGeneration: New Technologies and the Centrality of Pedagogy. *Harvard Educational Review*, 83(2), 300-319. <https://doi.org/10.17763/haer.83.2.w221368g1554u158>
- Pope, N. G. (2016). How the Time of Day Affects Productivity: Evidence from School Schedules. *The Review of Economics and Statistics*, 98(1), 1-11. https://doi.org/10.1162/REST_a_00525
- Prediger, S., Gravemeijer, K., & Confrey, J. (2015). Design research with a focus on learning processes: an overview on achievements and challenges. *ZDM*, 47(6), 877-891. <https://doi.org/10.1007/s11858-015-0722-3>
- Prestridge, S. (2012). The beliefs behind the teacher that influences their ICT practices. *Computers & Education*, 58, 449-458.
- Raspberry Pi Foundation. (2025a). *Enabling all young people to realise their full potential through computing*. raspberrypi.org
- Raspberry Pi Foundation. (2025b). *Hello World Magazine*. <https://www.raspberrypi.org/hello-world>
- Reeves, T. C., McKenney, S., & Herrington, J. (2011). Publishing and perishing: The critical importance of educational design research. *Australasian journal of educational technology*, 27, 55-65. <https://doi.org/10.14742/ajet.982>
- Riccomini, P. J., Smith, G. W., Hughes, E. M., & Fries, K. M. (2015). The Language of Mathematics: The Importance of Teaching and Learning Mathematical Vocabulary. *Reading & Writing Quarterly*, 31(3), 235-252. <https://doi.org/10.1080/10573569.2015.1030995>

- Rich, P. J., Browning, S. F., Perkins, M., Shoop, T., Yoshikawa, E., & Belikov, O. M. (2019). Coding in K-8: International Trends in Teaching Elementary/Primary Computing. *TechTrends*, 63(3), 311-329. <https://doi.org/10.1007/s11528-018-0295-4>
- Robbins, K., Senseman, D., & Pate, P. (2011). Teaching biologists to compute using data visualization. Proceedings of the 42nd ACM Technical Symposium on Computer Science Education,
- Robson, C. (2011). *Real World Research* (3rd ed.). Wiley.
- Rubin, H. J., & Rubin, I. S. (2005). *Qualitative Interviewing The Art of Hearing Data* (2nd ed.). SAGE.
- Sanford, J. F., & Naidu, J. T. (2016). Computational Thinking Concepts for Grade School. *Contemporary Issues in Education Research (CIER)*, 9(1), 23. <https://doi.org/10.19030/cier.v9i1.9547>
- Scornavacco, K., Kelly, M. R., & Boardman, A. (2022). Leading with Curricular Co-Design: An Exploration of Teacher Leadership Through the Co-Design Process. *Journal of Research on Leadership Education*, 17(4), 333-357. <https://doi.org/10.1177/19427751211017595>
- ScratchJr. (n.d.). *ScratchJr*. Retrieved 16/03/26 from <https://www.scratchjr.org/>
- Seidman, I. E. (1998). *Interviewing as qualitative research: a guide for researchers in education and the social sciences* (2nd ed.). Teachers College Press.
- Sen, S. (2023). Relations between preservice teachers' self-efficacy, computational thinking skills and metacognitive self-regulation. *European Journal of Psychology of Education*, 38(3), 1251-1269. <https://doi.org/10.1007/s10212-022-00651-8>
- Sentance, S. (2021). *Teaching computing in school: is K-12 research reaching classroom practice?* Proceedings of the 21st Koli Calling International Conference on Computing Education Research, Joensuu, Finland. <https://doi.org/10.1145/3488042.3491040>
- Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective [journal article]. *Education and Information Technologies*, 22(2), 469-495. <https://doi.org/10.1007/s10639-016-9482-0>
- Sentance, S., & Waite, J. (2018). Computing in the classroom: Tales from the chalkface. *it - Information Technology*, 60(2), 103-112. <https://doi.org/doi:10.1515/itit-2017-0014>
- Sentance, S., Waite, J., Hodges, S., MacLeod, E., & Yeomans, L. (2017). "Creating Cool Stuff": Pupils' Experience of the BBC micro:bit. Proc. 2017 ACM SIGCSE Tech. Symp. Computer Science education (Salem, Mass.),
- Sentance, S., Waite, J., Yeomans, L., & MacLeod, E. (2017). *Teaching with physical computing devices: the BBC micro:bit initiative* Proceedings of the 12th Workshop on Primary and Secondary Computing Education, Nijmegen, Netherlands. <https://doi.org/10.1145/3137065.3137083>
- Severance, S., Penuel, W. R., Sumner, T., & Leary, H. (2016). Organizing for Teacher Agency in Curricular Co-Design. *Journal of the Learning Sciences*, 25(4), 531-564. <https://doi.org/10.1080/10508406.2016.1207541>
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158. <https://doi.org/https://doi.org/10.1016/j.edurev.2017.09.003>
- Sims, S., Fletcher-Wood, H., Godfrey-Fausset, T., & Meliss, S. (2023). *Modelling evidence-based practice in initial teacher training: causal effects on teachers' skills, knowledge and self-efficacy*.

- Standards and Testing Agency. (2024a). *2025 optional key stage 1 tests guidance*. <https://www.gov.uk/government/publications/optional-key-stage-1-tests-guidance/2025-optional-key-stage-1-tests-guidance>
- Standards and Testing Agency. (2024b). *Reception baseline assessment: collection page*. <https://www.gov.uk/government/collections/reception-baseline>
Retrieved from <https://www.gov.uk/government/collections/reception-baseline>
- Sturman, L., Burge, B., Cook, R., & Weaving, H. (2012). *TIMSS 2011: Mathematics and Science Achievement in England*.
- Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, 26(1), 3-20. <https://doi.org/10.1007/s10798-015-9304-5>
- Swain, J. (2017). Ethical Issues and Considerations. In J. Swain (Ed.), *Designing Research in Education Concepts and Methodologies* (pp. 74-96). SAGE.
- Taub, R., Armoni, M., & Ben-Ari, M. (2012). CS Unplugged and Middle-School Students' Views, Attitudes, and Intentions Regarding CS. *ACM Transactions on Computing Education (TOCE)*, 12, 8:1-29. <https://doi.org/10.1145/2160547.2160551>
- Tedre, M., & Denning, P. J. (2016). *The long quest for computational thinking* Proceedings of the 16th Koli Calling International Conference on Computing Education Research, Koli, Finland. <https://doi.org/10.1145/2999541.2999542>
- The Design-Based Research Collective. (2003). Design-Based Research: An Emerging Paradigm for Educational Inquiry. *Educational Researcher*, 32(1), 5-8. <http://www.jstor.org.ezproxy.mmu.ac.uk/stable/3699927>
- The Open University. (2019). *Workplace learning with coaching and mentoring*. OpenLearn. Retrieved 21.09.25 from <https://www.open.edu/openlearn/money-business/workplace-learning-coaching-and-mentoring/content-section-3>
- The Royal Society. (2017). *After the reboot: computing education in UK schools*. T. R. Society. <https://royalsociety.org/-/media/events/2018/11/computing-education-1-year-on/after-the-reboot-report.pdf?la=en-GB&hash=346758F442E618BEFA28D1DACCC5B4A9>
- The Royal Society. (2019). Computing Education Snapshot: Policy briefing on teachers of computing [Policy briefing]. 1-7. Retrieved 05/12/2022, from <https://royalsociety.org/topics-policy/publications/2019/policy-briefing-on-teachers-of-computing/>
- Thompson, P., & Hall, C. (2017). *Place-Based Methods for Researching Schools*. Bloomsbury.
- Tondeur, J., van Braak, J., Ertmer, P. A., & Ottenbreit-Leftwich, A. (2017). Understanding the relationship between teachers' pedagogical beliefs and technology use in education: a systematic review of qualitative evidence. *Educational technology research and development*, 65(3), 555-577. <https://doi.org/10.1007/s11423-016-9481-2>
- Towhidnejad, M., Kestler, C., Jafer, S., & Nicholas, V. (2014). Introducing computational thinking through stealth teaching. IEEE Frontiers in Education Conference (FIE) Proceedings,
- Tran, D., & O'Connor, B. R. (2024). Teacher curriculum competence: how teachers act in curriculum making. *Journal of Curriculum Studies*, 56(1), 1-16. <https://doi.org/10.1080/00220272.2023.2271541>

- Tran, Y. (2019). Computational Thinking Equity in Elementary Classrooms: What Third-Grade Students Know and Can Do. *Journal of Educational Computing Research*, 57(1), 3-31. <https://doi.org/10.1177/0735633117743918>
- Vale, C., Roche, A., Cheeseman, J., Gervasoni, A., Livy, S., & Downton, A. (2023). The practices of middle leaders of mathematics: alignment of their goals and activities. *School Leadership & Management*, 1-23. <https://doi.org/10.1080/13632434.2023.2179982>
- Vermunt, J. D., Vrikki, M., Dudley, P., & Warwick, P. (2023). Relations between teacher learning patterns, personal and contextual factors, and learning outcomes in the context of Lesson Study. *Teaching and Teacher Education*, 133. <https://doi.org/10.1016/j.tate.2023.104295>
- Voogt, J., Fisser, P., Yadav, A., Good, J., & Mishra, P. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715-728. <https://doi.org/10.1007/s10639-015-9412-6>
- Voogt, J., Laferrière, T., Breuleux, A., Itow, R. C., Hickey, D. T., & McKenney, S. (2015). Collaborative design as a form of professional development. *Instructional Science*, 43(2), 259-282. <https://doi.org/10.1007/s11251-014-9340-7>
- Waite, J., Maton, K., Curzon, P., & Tuttiett, L. (2019, September). Unplugged Computing and Semantic Waves Analysing Crazy Characters,. Proceeding of UKICER – The UK and Ireland
- Computing Education Research Conference, University of Kent, Canterbury, UK.
- Wang, A., Yu, S., Wang, M., & Chen, L. (2024). Effects of the visual analytics of peer feedback on teachers' TPACK development in a lesson study. *Interactive learning environments*, 32(8), 4632-4651. <https://doi.org/10.1080/10494820.2023.2204354>
- Wang, F., & Hannafin, M. J. (2005). Design-Based Research and Technology-Enhanced Learning Environments. *Educational technology research and development*, 53(4), 5-23. <https://doi.org/10.1007/BF02504682>
- Wang, T., Olivier, D. F., & Chen, P. (2020). Creating individual and organizational readiness for change: conceptualization of system readiness for change in school education. *International Journal of Leadership in Education*, 1-25. <https://doi.org/10.1080/13603124.2020.1818131>
- 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. <https://doi.org/10.1007/s10956-015-9581-5>
- Wenner, J. A., & Campbell, T. (2017). The Theoretical and Empirical Basis of Teacher Leadership: A Review of the Literature. *Review of educational research*, 87(1), 134-171. <https://doi.org/10.3102/0034654316653478>
- Whyte, R. (2021). *Designing multimodal composition activities to support integrated primary programming and storytelling* (Publication Number 66886) [University of Nottingham]. Nottingham eTheses <https://eprints.nottingham.ac.uk/66886/>.
- Wiens, P. D., & Beck, J. S. (2022). But are they good teachers? examining who takes up teacher leadership and how their instruction differs from their peers. *School Leadership & Management*, 42(4), 381-401. <https://doi.org/10.1080/13632434.2022.2111413>
- Williams, R., Kamarulzaman, M. S., & Cuthrell, K. M. (2022). Reviewing Kennedy's Models Of CPD and the Relevance in TEL. *Asian journal of advances in research*, 17(1), 28-34.

- Wilson, A., & Moffat, D. C. (2010). *Evaluating Scratch to introduce younger schoolchildren to programming* 22nd Annual Workshop of the Psychology of Programming Interest Group, Leganés, Spain., Leganés, Spain.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2010). Computational Thinking: What and Why?
- Wolz, U., Stone, M., Pearson, K., Pulimood, S. M., & Switzer, M. (2011). Computational Thinking and Expository Writing in the Middle School. *ACM Transactions on Computing Education (TOCE)*, 11(2), 1-22.
<https://doi.org/10.1145/1993069.1993073>
- Wong, G. K. W. (2016). The behavioral intentions of Hong Kong primary teachers in adopting educational technology. *Educational technology research and development*, 64(2), 313-338.
<https://link.springer.com/article/10.1007/s11423-016-9426-9>
- Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational Thinking in Elementary and Secondary Teacher Education. *ACM Transactions on Computing Education (TOCE)*, 14(1), 1-16.
<https://doi.org/10.1145/2576872>
- Yadav, A., Stephenson, C., & Hong, H. (2017). Computational Thinking for Teacher Education. *Communications of the ACM*, 60(4), 55-62.
- Yin, R. K. (2018). *Case Study Research and Applications* (6th ed.). SAGE.
- Zhou, N., Nguyen, H., Fischer, C., Richardson, D., & Warschauer, M. (2020). High School Teachers' Self-efficacy in Teaching Computer Science. *ACM transactions on computing education*, 20(3), 1-18.
<https://doi.org/10.1145/3410631>

APPENDICES

Appendix 1: Semi-structured interview questions for the head teacher and computing co-ordinator, adapted from Israel et al (2015)

Questions for the headteacher

1. How has computing changed in light of the New curriculum? (Priority as a probe)
2. What is your view of computing across the school?
3. What has surprised you most about computing at your school? Have there been any specific opportunities? Probe for answers related to students and teachers.
4. Have there been any challenges/ barriers to implementing computing across the school? Were they things you expected?
5. What kind of feedback have you received from the teachers who have implemented computing?
6. Have any teachers been hesitant to teach computing? Why?
7. Have you had any interesting responses from parents related to the introduction of computing to the curriculum?

Questions for the computing co-ordinator

1. How did you become the computing coordinator in the school?
2. How do you see your role as computing co-ordinator?
3. How has your role changed in light of the new curriculum?

4. What resources do teachers currently use when teaching computing?
(schemes of work/ physical resources)
5. How do you see computing developing across the school? Discrete
subject/ cross-curricular...
6. What do you hope computing will look like next year?
7. What has surprised you about computing at ____? Have any exciting
opportunities have developed? Teachers/ students
8. What challenges/ barriers do you see regarding computing (sub
knowledge/ resourcing...)
9. What sort of feedback have you had from teachers regarding the new
computing curriculum?
10. Have any areas of the computing curriculum caused more challenges
than others or caused more opportunities?

Appendix 2: Class teacher interviews

1. How does the thought of teaching computing make you feel?

2. What factors contribute to this?
3. How does computing compare to other subjects in the curriculum?
4. How much teaching of computing do you do in a typical week? Is this usually discrete lessons or as part of other subjects? Are there any occasions when you teach more, or less computing?
 - 5a. Are there any areas of the computing curriculum your prefer teaching? Why
 - 5b. Are there any areas you are less confident teaching? Why?
6. Have you had any successes/ surprises when teaching computing? Can you describe a lesson that went well.
7. Have there been any barriers/ challenges when teaching computing? Can you describe a computing lesson that didn't go as planned and why?
8. Does anyone else in your classroom contribute to the teaching of computing? If so how?
9. Did you teach the old computing curriculum? If so how does this compare?
10. What key stage do you teach in? Would you feel confident to teach computing in another key stage?
11. What was your own experience of computing at school? What about during teacher training?
12. What support would help you in the teaching of computing?

13. Is there anything else you'd like to add about the computing curriculum or teaching computing in general?

Appendix 3: Lesson plan for teaching micro:bits

Introduction to micro:bits session plan for UKS2

Learning goals

- Can I explain what a micro:bit is?
- Can I upload a program to the micro:bit?
- Can I use a block editor to write an algorithm that controls the LEDs on a micro:bit?

Activities

Discussion (5mins)

1. Display the keywords ‘algorithm’ and ‘program.’ Give the children a minute to talk to a talk partner. Assess prior knowledge and give feedback to the group.

Meet the micro:bit (10mins)

2. Show the children a micro:bit and simultaneously display a picture on the screen. Discuss what the micro:bit is and what it can do; explain that they will use them in secondary school but that we have a set for them to use here in school.
3. Focus on the LEDs and show an image on the screen, pointing out the coordinates

Programming the micro:bit (15mins)

4. Bring up the JavaScript Blocks Editor on the screen <https://makecode.microbit.org/>
<https://makecode.microbit.org/>

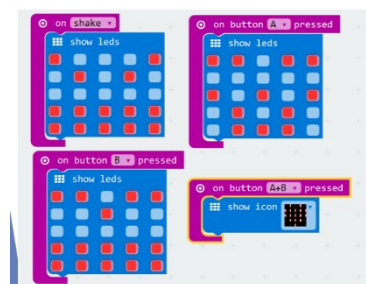
Explain to the children that they will use a form of block programming similar to scratch, allowing them to upload a program to the micro:bit. Explain that they will create a Tamagotchi from their micro:bit that will show a different face in response to a different input.

Java Blocks Code Editor



5. Hand out laptops and blank Tamagotchi design sheets—children to design their face designs. Talk through opening the micro:bit site, clicking on ‘new project’, and the different inputs. Allow the children time to explore the block program, then create their basic micro:bit program.

Tamagotchi!



Uploading program to the micro:bit (10mins)

6. Bring the class back together. Demonstrate how to download their program, then connect the micro:bit to the laptops¹⁸ and upload the program. Allow time for testing.

Plenary (10mins)

7. Show the key skills of computational thinkers. Discuss each term. Do the children feel that they demonstrated them?
8. Look at the original LO as I can statements. Do the children think they have met them?

Key skills!

What do computational thinkers do?!

- ▶ Creating
- ▶ Tinkering
- ▶ Debugging
- ▶ Persevering
- ▶ Collaborating



¹⁸ A different procedure is needed for downloading and uploading a program if using iPads.

Appendix 4: CPD Lesson Plan

CPD- Computational thinking in the primary school. Watermill School 25.09.19 60mins	
Learning Intentions - Consider key computational thinking terminology in a meaningful context! - Plan some cross-curricular activities to teach computational thinking.	Resources Required - Flip chart paper - Pens - Copies of the autumn term curriculum document for each class.

Teaching sequence		Purpose
2	Slide Prior knowledge What is computational thinking? In pairs on large sheets of paper – what do we know about CT so far? Divide into 3: keywords, concepts, and skills.	Gather participants' prior understanding of the key concepts

Slides 3 and 4	• A summary of findings so far (sum up what I have found)... - What has come out of your interviews/ observations etc.? - Time, Use of Language, 'Things' not working • LI for today's session...	Background to the project. Explain purpose. Engage with the project.
Slide 5	• Language – why it is important and how not to turn it into 'fronted adverbials!' • Activity- match the vocab. • What does this language look like at WM? • Progression of language- AL's medium-term plan for computing- where are the computational thinking bits?	WM focus on language; ensure that this does not become the overriding focus of teaching/ thinking.

6-12 Slides	• Decomposition and Pattern recognition maths: clean your teeth • Abstraction- you're only allowed 4 stages – which ones are key? • How else could we teach this- literacy- decompose a story/ story maps- things we do already. • Explain the story to... ‘	Demonstrate activities that can help to reinforce the key activities that don't involve using computers. Meaningful context for the language- not learning for the sake of it.
---------------------------	---	--

Slides 13 and 14	• Time pressures/ Cross-curricular learning – where can we fit this into the curriculum? • Resource pressures- lucky with what you have at WM but can also adapt what we have/ what we're teaching already for unplugged lessons. • Block overviews—What is on the curriculum for this Block? Where could we fit in CT? • E.G. Team 1- Exploring our school. Plan a route for Bee-Bot around the school. • Lesson planning	Teachers to look at their curriculum plans and identify opportunities for teaching CT
	Follow up • Plan lessons in pairs/ triads? Observe/ get involved in the lessons.	

Appendix 5: Ethics documents

Participant teacher's consent form

Project title: Primary Computing Three Years On; How can we change the teaching of computational thinking within a primary school?

Researcher's name: Gemma Rolfe

Supervisor's name: Shaaron Ainsworth

- I have read the Information Sheet and the nature of the research project has been explained to me. I understand the purpose of the research project and my involvement in it.
- I agree to be interviewed and/ or answer a questionnaire as part of the research
- I am happy for my interview to be video/audio recorded
- I agree to be observed as part of the project and understand that this will not, in any way, be used for performance management purposes.
- I understand that if any of my words are used in reports or presentations they will not be attributed to me.
- I understand that I can withdraw from the research project at any stage and that this will not affect my status now or in the future. I will still be able to be involved in the school's CPD.
- I understand that should I wish to withdraw from the intervention any personal data collected will be removed from the study and destroyed.
- I understand that data will be stored on a secure password-protected University server for 7 years, after which it will be destroyed, and will only be accessible by the researcher, Gemma Rolfe and her supervisor, Shaaron Ainsworth.
- I understand that I may contact the researcher or supervisor if I require further information about the research, and that I may contact the Research Ethics

Coordinator of the School of Education, University of Nottingham, if I wish to make a complaint relating to my involvement in the research.

Name _____

Signed _____ Date _____

Researcher's name _____ Signed _____

Contact details

Researcher: Gemma Rolfe gemma.rolfe@nottingham.ac.uk

Supervisor: Shaaron Ainsworth

shaaron.ainsworth@nottingham.ac.uk 0115 846 7671

School of Education Research Ethics Coordinator:

educationresearchethics@nottingham.ac.uk

Participant Teacher's information sheet

Understanding the teaching of computing within a primary school

Dear teachers,

My name is Gemma Rolfe and I am a doctoral student at the School of Education at the University of Nottingham. I am also a lecturer at Manchester Metropolitan University, in the Faculty of Education. I am currently researching how the new 2014 Computing Curriculum has impacted on the teaching of computing within primary schools. I am particularly interested in looking at whether the new curriculum has changed what is currently taught and what challenges teachers face in helping children learn the new curriculum. I will then be looking to develop professional development materials, with teachers in your school, which can be used to help other teachers. I am specifically focussing on computational thinking; a series of concepts and skills, including coding or programming, that are regarded as important 21st century skills and are part of the curriculum from the beginning of key stage 1.

The first phase of my research involves gaining a picture of what is currently taught in a school; I am looking to celebrate successes but also to investigate any challenges you might have faced as a school. I am writing to ask if you would be interested in collaborating with me on this research project. This would involve me interviewing key members of staff, including the head-teacher and computing co-ordinator, seeking your views regarding computing through a questionnaire, and observing the teaching of computing within the school. Any observations that take place will be entirely for the

purposes of my research and would not, in any way, be related to performance management. The observations will not directly involve the children in your classes and will not impact on the flow of the lessons. As part of the study I will be providing CPD for all teachers, and interested staff, within the school- if you choose not to participate in the study you will still be entitled to take part in this.

The proposed research project has been approved by the School of Education Ethics Committee at the University of Nottingham, in compliance with the University's Code of Research Conduct and Research ethics and follows the ethical protocols outlined by the British Educational Research Association. This means that the school's and teachers' identities will be anonymised from the outset and all materials relating to participants in the study would be confidential. All data collected will be stored for 7 years following any publication of the research, after which it will be destroyed. These data are only available to my supervisors and myself. All participants have the right to withdraw themselves from all, or part, of the study at any point. The results of the project will be provided to the school. Once I have analysed the data I will use this for my EdD thesis and for publication. I hope that the findings will help to improve the picture of how the new curriculum is being taught and will provide practical help for teachers faced with teaching something new.

I have a clean DBS background check and taught in Cheshire schools for 10 years before starting work at Manchester Met; as a lecturer at MMU I regularly carry out observations of teaching. The observations will in no way

impact on the general flow of the lesson and will not involve the children in any way.

If you are happy with the nature and purpose of the research project, and agree to participating, then please complete and return the attached form by _____. If you have any questions, or require more information about the project, please contact me, my supervisor or the ethics committee, using the contact details attached.

Yours faithfully,

Gemma Rolfe

School of Education, University of Nottingham

Contact details

Researcher: Gemma Rolfe gemma.rolfe@nottingham.ac.uk

Supervisor: Shaaron Ainsworth

shaaron.ainsworth@nottingham.ac.uk 0115 846 7671

School of Education Research Ethics Coordinator:

educationresearchethics@nottingham.ac.uk

Parents' Information Sheet

Information sheet for Primary Computing Three Years On: How can we change the teaching of computational thinking within a primary school?

Dear Parents,

My name is Gemma Rolfe and I am a doctoral student at the Learning Sciences Research Institute, School of Education at the University of Nottingham; I am also a lecturer at Manchester Metropolitan University, in the Faculty of Education. I am currently researching how the new 2014 Computing Curriculum has impacted on the teaching of computing within primary schools. I am particularly interested in looking at whether the new curriculum has changed what is currently taught and whether there have been any challenges to this; I will then be looking to develop CPD, with teachers in your school, which can be used to help other teachers. I am specifically focussing on computational thinking; a series of concepts and skills, including coding or programming, that are regarded as important 21st century skills and are part of the curriculum from the beginning of key stage 1.

The first phase of my research involves gaining a picture of what is currently taught in a school; I am looking to celebrate successes but also to investigate any challenges that the school might have faced such as new subject knowledge and resources required. I will therefore be working with the school to observe teachers teaching computing and finding out their views of the new curriculum. Whilst your children will obviously be in the lessons whilst computing is being taught, I will not be working with them directly or

gathering their views at this time. The observations will in no way impact on the general flow of the lesson and will not involve the children in any way.

The proposed research project has been approved by the University of Nottingham Ethics committee and follows the ethical protocols outlined by the British Educational Research Association. This means that the school's and teachers' identities will be anonymised from the outset and all materials relating to participants in the study would be confidential. Following the completion of the research, all data collected will be stored for 7 years, after which it will be destroyed, and only available to my supervisors and me. The results of the project will be provided to the school. Once I have analysed the data I will use this for my EdD thesis and for publication. I hope that the findings will help to improve the picture of how the new curriculum is being taught and will provide practical help for teachers faced with teaching something new.

I have a clean DBS background check and taught in Cheshire schools for 10 years before starting work at Manchester Met; as a lecturer at MMU I regularly carry out observations of teaching.

If you have any questions, or require more information about the project, please contact me, my supervisor or the ethics committee, using the contact details attached.

Yours faithfully,

Gemma Rolfe

Learning Science Research Institute,
School of Education, University of Nottingham

Contact details

Researcher: Gemma Rolfe gemma.rolfe@nottingham.ac.uk

Supervisor: Shaaron Ainsworth
shaaron.ainsworth@nottingham.ac.uk 0115 846 7671

School of Education Research Ethics Coordinator:
educationresearchethics@nottingham.ac.uk

Participants GDPR Information Sheet

Research participant privacy notice

Privacy Information for Research Participants

For information about the University's obligations with respect to your data, who you can get in touch with and your rights as a data subject, please visit: <https://www.nottingham.ac.uk/utilities/privacy.aspx>.

Why we collect your personal data

We collect personal data under the terms of the University's Royal Charter in our capacity as a teaching and research body to advance education and learning. Specific purposes for data collection on this occasion are to understand teachers experiences of the new computing curriculum within schools.

Legal basis for processing your personal data under GDPR

The legal basis for processing your personal data on this occasion is Article 6(1a) consent of the data subject.

How long we keep your data

The University may store your data for up to 25 years and for a period of no less than 7 years after the research project finishes. The researchers who gathered or processed the data may also store the data indefinitely and reuse it in future research. Measures to safeguard your stored data include storing data in a locked work desk and on the university server. All participants will be given a pseudonym which will only be known by the researcher and supervisor.

Who we share your data with

Extracts of your data may be disclosed in published works that are posted online for use by the scientific community. Your data may also be stored indefinitely on external data repositories (e.g., the UK Data Archive) and be further processed for archiving purposes in the public interest, or for historical, scientific or statistical purposes. It may also move with the researcher who collected your data to another institution in the future.