

# Comparing K-5 teachers' reported use of design in teaching programming and planning in teaching writing

Jane Waite

Queen Mary University of London  
j.l.waite@qmul.ac.uk

William Marsh

Queen Mary University of London  
d.w.r.marsh@qmul.ac.uk

Paul Curzon

Queen Mary University of London  
p.curzon@qmul.ac.uk

Sue Sentance

King's College London  
sue.sentance@kcl.ac.uk

## ABSTRACT

K-5 teachers teach a range of subjects & develop generic teaching skills; when starting to teach computing, particularly programming, practitioners may not realise that they can draw on these other skills to support their teaching. In a small study of K-5 teachers, potential synergies were suggested between using *planning* in the teaching of writing and *design* in the teaching of programming. In this paper, we explore these synergies by surveying a wider group of teachers (n=207) on their uses of planning and design. Teachers reported the usefulness of planning for writing and design for programming as equally important. However, there were significant differences in their uses. The majority saw planning as essential in writing & put this into practice in their teaching. For example, they demonstrated the creation of plans, expected students to annotate plans, required students to refer to plans when writing and used plans to differentiate. By contrast, these uses were implemented less frequently in programming tasks. We also report on differences in the confidence of male & female respondents, & between generalists (who teach programming & writing) & specialists (who do not teach writing). For example, females were more confident to teach writing than programming, with males vice versa. Having revealed opportunities for reuse of successful techniques used in teaching writing for the teaching of programming we recommend further work is needed to explore this transfer of pedagogical knowledge.

## CCS CONCEPTS

• **Social and professional topics** → **K-12 education**;

## KEYWORDS

K-5 Computing Education, Teachers, Design, Programming

### ACM Reference Format:

Jane Waite, Paul Curzon, William Marsh, and Sue Sentance. 2018. Comparing K-5 teachers' reported use of design in teaching programming and planning in teaching writing. In *Proceedings of the 13th Workshop in Primary and*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

WiPSCE '18, October 4–6, 2018, Potsdam, Germany

© 2018 Copyright held by the owner/author(s). Publication rights licensed to the Association for Computing Machinery.

ACM ISBN 978-1-4503-6588-8/18/10...\$15.00

<https://doi.org/10.1145/3265757.3265761>

*Secondary Computing Education (WiPSCE '18)*, October 4–6, 2018, Potsdam, Germany. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3265757.3265761>

## 1 INTRODUCTION

Schools have been required in England to teach K-5 learners how to program since 2014 [7] including asking learners to 'Design, write and debug programs'. However, the extent they are using design and what its impact is, is unclear. In a review of international resources for computing teachers, design has been cited as lacking [8]. A study matching goals to be taught against research related to those goals cited design as the most unmatched goal [24]. Recent research has investigated approaches for incorporating design, including: learning paths incorporating exploration and design [29]; design in blended approaches [10]; design of augmented games [14]; use of industry approaches tailored to education such as the engineering design process [3] and agile methods [15]; and emphasizing self-awareness of the process of design [17]. Despite a promising start, this work is limited in scale, length of study and has not been undertaken in the context of the early introduction of Computing in schools, such as in the English Curriculum. Compounding the lack of supporting resources and limited targeted research, few K-5 teachers have been taught programming or design. Despite initial professional development efforts, the need for further teacher support and training has been called for [31].

K-5 teachers teach a range of subjects, and in so doing develop pedagogical content knowledge across differing contexts [30]. Writing is a core subject taught in K-5 classrooms. Planning, as used in teaching writing, has potential synergies with design. This raises the possibility of transferring common instructional techniques from the teaching of writing to the teaching of programming [35].

Planning in writing has a high impact on the improvement of learners' writing as it supports self-regulation during revising and editing compositions [9, 12]. The plan helps learners focus on one part of a complex activity at a time, reducing cognitive load. It separates out the task of gathering ideas and developing an initial overall piece from the job of then implementing this as written text. Learners can annotate their plan with useful words and phrases, without contending with how the fragments might be joined to create larger structures. At the end of writing the plan can be used to check that all intended parts are included, checking for completeness and cohesion. Such strategies have been claimed to support struggling writers, providing improved attainment through

pupils being able to understand the processes involved and focus on just what is required next [27].

Transfer of pedagogical expertise is a complex area [22]. It has been considered in the light of personal experience in physical education [20] and in transfer across physics topics [18]. Whether teachers are utilizing planning or design in K-5 classes, and what the relative comparative use is, has not been explored at this point in time, with programming now a mandatory requirement in K-5 English classrooms [7], nor has whether teachers can, or will, transfer pedagogical expertise across these contexts, or whether this is a useful or effective approach.

This study is part of a larger programme investigating the use of abstraction in the teaching and learning of K-5 programming in schools. In our first qualitative study, we interviewed 5 teachers and 50 pupils to explore their use of one form of abstraction in action, that of design in programming projects and planning in other subjects [35]. Referencing a level of abstraction framework used with older learners and their teachers [1, 23, 32] we renamed the object or algorithm level the ‘design level’ for K-5 [36]. Using this framework as a lens, we reviewed teachers’ uses and understanding of the levels of abstraction framework, particularly the design level [35]. We proposed that there were possible synergies between planning to teach writing and design to teach programming but suggested further investigation to discover if interview results were generalizable across a wider population.

The contribution of this paper is to acquire underpinning data that shows the extent to which planning and design are used in K-5 classrooms as well as their perceived usefulness. This provides a basis for future exploration of the potential transfer of K-5 teachers’ experience of using planning when teaching writing to using design when teaching programming.

## 2 STUDY AIMS

To address the question as to whether a wider population of teachers used planning and design in the ways suggested by our first study, we created a survey asking teachers directly if they used planning and design in the ways it suggested. In this paper, we focus on the survey questions related to confidence to teach writing and programming; usefulness of planning & design; and 5 core specific uses of planning and design. The related three main research questions are RQ1: Are teachers’ **confidence** to teach programming & writing the same?; RQ2: Are there relationships between **usefulness & use** of design & planning?; RQ3: Do teachers use design and planning in the same way for **specific uses**?

### 2.1 Methodology

Surveys are a way of eliciting information from a wide group of participants where interviews are not practicable on a large scale [5]. We used questions based on our earlier findings to discover whether ideas reported by teachers in interviews were more widely held.

**2.1.1 Participants.** Participants were recruited from four sources: a general survey advertised on social media, open to any teacher; a pre-course survey of teachers attending a short 1 hour workshop advertised to a local area; a pre-course survey of teachers attending a long 3 day course for more experienced K-5 computing teachers

on teaching programming in general (called ‘Diving Deep into Primary Programming’); and a pre-course survey of teachers enrolled on the BCS Certificate Scratch course. Data was gathered from June 2017 to April 2018. This accommodated four cohorts of the Diving Deep course and three of the BCS certificate. The general survey was advertised on Twitter with links to general primary teacher groups as well as computing groups.

**Generalists & Specialists** One aspect of the study was to compare teachers who taught programming and writing (who we call *generalists*) to those who did not teach writing but did teach programming (who we call *specialists*).

**2.1.2 Survey design.** The survey was trialled with several teachers who gave feedback on the questions, allowing them to be improved before the full study. Ethics procedures were followed throughout, with approval obtained from the university’s ethics panel. The survey used Likert scales to allow participants to rank their responses. These were combined with open questions to allow extra input. Qualitative data from open questions is not analysed here but will be reported subsequently. The survey was divided into sections, only those parts which we report on here are described.

**Programming Section** The 1st question was mandatory and asked, ‘Do you teach programming?’ If the participant answered ‘No’, they were moved on to the Writing Section. Otherwise, they were asked questions related to teaching programming with the 1st question: ‘How confident are you teaching programming?’ Questions related to the use of design were asked, these were repeated in the Writing Section. The questions related directly to the results from our early study of interviewed teachers reported use of design. One such question was ‘For programming projects, do you use the designs created by pupils to work out what to teach next?’

**Writing Section** The 1st question in the writing section was mandatory and asked, ‘Do you teach writing? (By writing we mean literacy or English lessons.)’ If the participant answered ‘No’, they were moved on to the end section. Otherwise, they were asked questions related to teaching writing with the 1st question: ‘How confident are you teaching writing?’ In the writing section, the questions mirrored those in the Programming Section. For example, ‘For substantial writing tasks, do you use the plans created by pupils to work out what to teach next?’

**End Section** The end section contained general questions on gender, ages taught and length of time teaching. All teachers were asked whether they thought design was useful and what drawbacks and advantages there were for the use of design in programming projects. The same questions were asked of planning in substantial writing tasks. The survey is available for reuse by contacting the lead researcher.

**2.1.3 Data analysis approach.** After the survey was completed data was analysed using SPSS. Our results are predominantly non-parametric data, therefore, we have used the Mann Whitney U test to compare two independent groups such as male and female or generalist and specialist [5] and the Kruskal-Wallis test to investigate differences across 3 or more independent groups such as comparing Likert scale responses [5]. For both these tests, cross-tabulations were used to investigate any statistical significance and the effect size calculated. For the Kruskal-Wallis tests, we report the  $\chi^2$  statistic. We have not completed post-hoc analysis through

**Table 1: Overview of participants male or female & specialist or generalist by data source**

Data Source	Male	Female	No Answer	Specialist	Generalist	Total
General Survey	36%(46)	63%(82)	1% (1)	49%(63)	51%(66)	62%(129)
Diving Deep	55%(28)	43%(22)	2%(1)	65%(33)	35%(18)	25%(51)
Design Wkshop	27%(4)	73%(11)	0	20%(3)	80%(12)	7%(15)
BCS Cert.	16%(2)	83%(10)	0	58%(7)	42%(5)	6%(12)
Totals	39%(80)	60%(125)	1% (2)	51%(106)	49% (101)	207

**Table 2: Who teaches who?**

Band	Male	Female	No Ans	Specialist	Generalist	Total
Primary	37%(71)	62%(118)	1%(1)	48%(91)	52%(99)	92%(190)
Secondary	43%(15)	57%(20)	0%(0)	80%(28)	20%(7)	17%(35)
PostFE	49%(17)	49%(17)	2%(1)	86%(30)	14%(5)	17%(35)

the Bonferroni-correction for pairwise comparison of categories as our focus has been on establishing overall trends and differences. To grade effect sizes, we used Cohen's classification: if  $r$  is 0.1 to 0.3 there is small effect, if  $r$  is 0.3 to 0.5 there is moderate effect and 0.5 and above is large effect. For paired dependent testing, such as comparing each teacher's responses to the same detailed question about design and planning, the Wilcoxon Signed-Rank test [19] was used. We show the number of cases tied and the changing 'up' or 'down' of ranked responses to report on the underlying data.

### 3 RESULTS

#### 3.1 Participant Background

207 participants completed the survey. Females were the most represented gender at 60% ( $n=125$ ): see Table 1. The general survey provided the largest subgroup of participants at 62% ( $n=129$ ). General survey females ( $n=82$ ) made the largest data source gender subset at 40% of overall population. (Two respondents chose not to answer the gender question: one from the General Survey and one from the Diving Deep data source. They recorded confidence of levels 4 and 5 and were both specialists.)

49% of respondents were generalists ( $n=101$ ) and 51% specialists ( $n=106$ ), as shown in Table 1. 38% ( $n=40$ ) of the specialists were male and 60% ( $n=64$ ) female. 40% ( $n=40$ ) of the generalists were male and 60% ( $n=61$ ) female.

Teachers were asked what age groups they taught such as: Early Years (ages 4-5); Key Stage 1 (ages 5-7). We summarise these groups into phases: Primary (ages 4-11); Secondary (ages 12-18) and Post Further Education (University and Adult) called Post FE (see Table 2). Teachers could select more than one age group; e.g., they could record they taught in primary & secondary. Hence, phases are not mutually exclusive. The vast majority of our sample taught primary only (72%) or taught primary and some other age group (92%).

**Table 3: Confidence to teach programming by gender**

Gender	1	2	3	4	5	Total
Male	39%(31)	27%(22)	19%(15)	14%(11)	1%(1)	39%(80)
Female	19%(24)	27%(33)	28%(35)	20%(25)	6%(8)	60%(125)
No answer	0	0	50%(1)	50%(1)	0	1%(2)
Total	27%(55)	27%(55)	24%(51)	18%(37)	4%(9)	207

**Table 4: Confidence to teach writing by gender**

Gender	1	2	3	4	5	Total
Male	22%(9)	45%(18)	15%(6)	13%(5)	5%(2)	40%(40)
Female	43%(26)	36%(22)	8%(5)	11%(7)	2%(1)	60%(61)
Total	35%(35)	39%(40)	11%(11)	12%(12)	3%(3)	101

**Table 5: Confidence to teach programming by specialist/generalist**

Group	1	2	3	4	5	Total
Specialists	30%(32)	36%(38)	17%(18)	13%(14)	4%(4)	106
Generalists	22%(23)	17%(17)	33%(33)	23%(23)	5%(5)	101

#### 3.2 Confidence

**3.2.1 Confidence to teach programming / writing by gender.** Participants were asked, 'How confident are you teaching programming?' based on a scale of 1 (Very Confident) to 5 (Not at all confident). Males ( $n=80$ ) reported a higher degree of confidence than females ( $n=125$ ) to teach programming ( $n=205$ ,  $U=3651$ ,  $r=-0.234$ ,  $p=0.001$ ), with moderate effect size (see Table 3). By contrast, females ( $n=61$ ) reported a higher degree of confidence to teach writing ( $n=101$ ,  $U=950$ ,  $r=-0.197$ ,  $p=0.048$ ) than males ( $n=40$ ), with small effect size (see Table 4).

**3.2.2 Confidence to teach programming by generalist or specialist & gender.** Specialists ( $n=106$ ) reported a higher confidence to teach programming ( $n=207$ ,  $U=4159$ ,  $r=-0.1985$ ,  $p=0.004$ ) than generalists ( $n=101$ ) with small effect size (see Table 5). Within the specialist group, female ( $n=60$ ) specialists were no more or less confident than their male specialist colleagues ( $n=40$ ), with small effect size.

A paired comparison of confidence for generalists ( $n=101$ ) showed they were more confident to teach writing than programming ( $Z=-3.909$ ,  $r=.388$ ,  $p<0.000$ ). 26% ( $n=26$ ) tied their confidence across the two subjects, 52% ( $n=53$ ) reported higher confidence in writing than programming and 22% ( $n=22$ ) vice versa.

**3.2.3 Length of time teaching subjects & confidence.** Teachers reported marked differences in the length of time teaching programming to teaching writing. 23% of teachers reported having taught programming for more than 5 years compared to nearly three times this (64%) having taught writing for this same period.

Teachers broadly reported a higher degree of confidence to teach programming the longer they had taught programming ( $n=207$ ,  $\chi^2=25.01$ ,  $d.f.=4$ ,  $p<.000$ ), as shown in Figure 1.

Most teachers (87%) had taught writing for more than three years, and most (79%) were either 'Very confident' to teach writing or one rating adjacent to this ( $\chi^2=16.285$ ,  $d.f.=4$ ,  $p=0.003$ ). However,

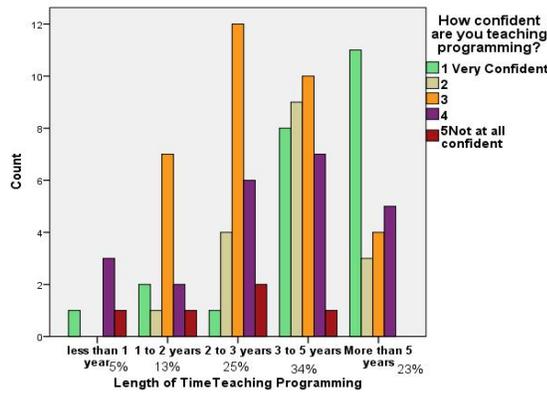


Figure 1: Bar chart of confidence to teach programming by length of time teaching programming

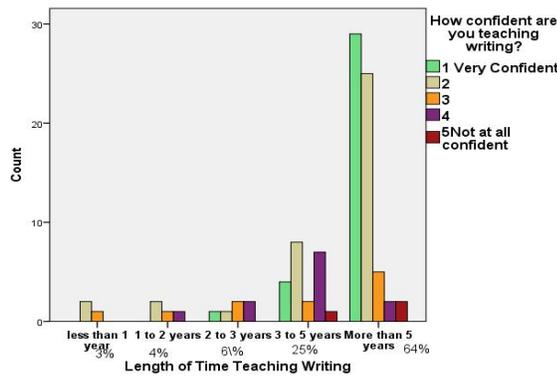


Figure 2: Bar chart of confidence to teach writing by length of time teaching writing

this pattern of matched experience and confidence did not hold, for shorter lengths of time teaching writing, as shown in Figure 2, where confidence was far more variable.

### 3.3 Usefulness of design & planning

We asked teachers ‘Do you think design in programming projects is useful?’ and ‘Do you think planning in substantial writing tasks is useful?’ These questions were asked of all teachers irrespective of whether they said they taught writing or not. Teachers’ responses were for **design** (n=207) mode = 1 (‘Essential’), mean = 1.7 and for **planning** (n=207) mode = 1 (‘Always’), mean = 1.8. As shown in Table 6, the majority of teachers reported design (82%) & planning (78%) as ‘Very Useful’ or ‘Essential’ with no teachers reporting design as ‘Not useful’. Only very few % (2%) reported this for planning.

**3.3.1 Usefulness & gender.** There was no statistically significant difference between male (n=80) and female (n=125) teachers’ reported usefulness of planning (n=205, U=4545.5, r=0.0832, p=0.234) and similarly no significant difference for male (n=80) and female (n=125) usefulness of design (n=205, U=4833.5, r=.03094, p=.658).

Table 6: Usefulness of design & usefulness of planning

Usefulness	Design			Planning		
	%	n	Accum%	%	n	Accum%
Essential	53%	110	53%	51%	105	51%
Very Useful	29%	59	82%	27%	56	78%
Somewhat Useful	15%	32	97%	16%	33	94%
A little Useful	3%	6	100%	4%	9	98%
Not Useful	0%	0	100%	2%	4	100%

Table 7: Cross tabulation of usefulness by specialist/generalist

Group	Essential	Very Useful	Somewhat Useful	A little Useful	Not useful
Design Generalist	44%(45)	28%(28)	24%(24)	4%(4)	0%(0)
Design Specialist	61%(65)	29%(31)	8%(8)	2%(2)	0%(0)
Plan Generalist	60%(61)	24%(24)	12%(12)	2%(2)	2%(2)
Plan Specialist	41%(44)	30%(32)	20%(21)	7%(7)	2%(2)

Table 8: Paired usefulness of design & planning

Group	Tied	Planning more useful	Design more useful
Generalists	57% (57)	31% (31)	13% (13)
Specialists	44% (46)	13% (14)	43% (46)

**3.3.2 Usefulness & generalist or specialist.** By contrast, there was a significant difference, between generalists (n=106) and specialists (n=101) for the usefulness of both planning and design. Generalists reported a higher usefulness of planning than specialists (U=4231, r=.1968, p=0.005), whereas specialists reported higher usefulness of design than generalists (U=4169, r=.2105, p=0.002), both with small effect size (see Tables 7).

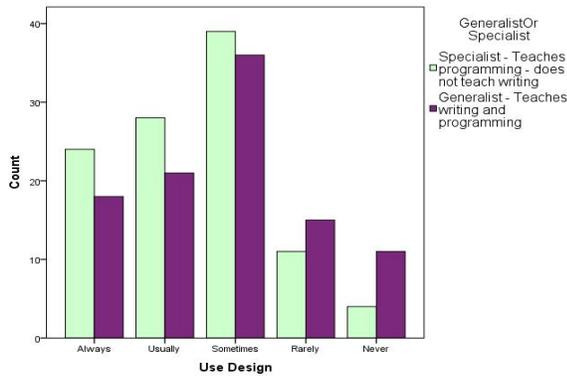
**3.3.3 Usefulness of design & usefulness of planning (paired).** There was no statistical difference of teachers’ (n=207) views of usefulness for design & usefulness of planning (Z=-1.537, r=-.106, p=0.124). Similarly, there was no significant difference for males (n=80, Z=-1.348, r=0.151, p=0.178) or females (n=125, Z=-1.082, r=0.097, p=0.279). Table 8 shows paired preference for generalists and specialists. Generalists (n=101) recorded planning as more useful than design (Z=-3.044, r=0.302, p=0.002), whereas, specialists (n=106) voted the opposite way (Z=-4.076, r=0.396, p<000), both with moderate effect size.

### 3.4 Use of design & use of planning

Teachers were asked ‘When teaching programming, do you require your pupils to consider or use a design at some point?’ By design, we mean a high-level plan of what the program is intended to do. A design could be verbal, written or drawn, informal or formal.’ Teachers who also reported they taught writing (generalists) were asked ‘When teaching writing of substantial pieces of work, do you require pupils to use a high-level plan?’ (By substantial we mean a task that will take several days to write, or that is more

**Table 9: Use of design & use of planning**

Use	Design			Planning		
	%	n	Accum%	%	n	Accum%
Always	20%	42	20%	57%	58	57%
Usually	24%	49	44%	31%	31	88%
Sometimes	36%	75	80%	9%	9	97%
Rarely	13%	26	93%	2%	2	99%
Never	7%	15	100%	1%	1	100%



**Figure 3: Bar chart of the reported use of design by specialists and generalists**

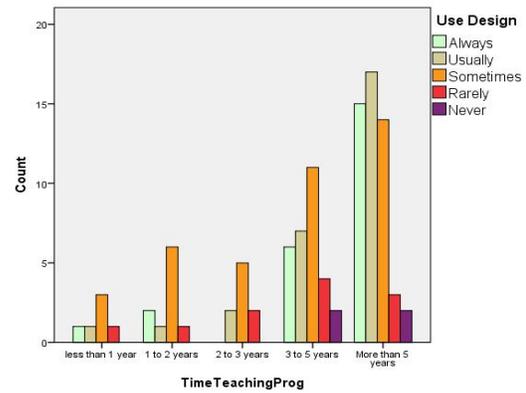
than a few paragraphs, such as a developed story, or non-fiction piece.) For both questions, they could answer ‘Always’, ‘Usually’, ‘Sometimes’, ‘Rarely’ or ‘Never’. Teachers’ responses were **design**: (n=207) mode = 3 (‘**Sometimes**’) mean= 2.6; **planning**: (n=101) mode = 1 (‘**Always**’) mean = 1.6. As shown in Table 9, 88% of our teachers ‘Always’ or ‘Usually’ required their pupils to use a plan at some point during writing whereas half of this number only 44% of teachers ‘Always’ or ‘Usually’ required pupils to consider or use design at some point during programming.

**3.4.1 Use & gender.** There was no statistically significant difference between male (n=40) and female (n=61) teachers’ reported use of planning (U=1242.5, r=0.0176, p=0.860) and similarly no significant difference for males and females for reported use of design (U=5713.5, r=-0.1247, p=.074).

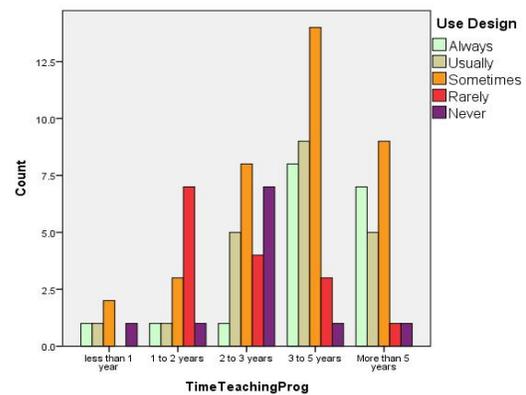
**3.4.2 Use of design & generalist or specialist.** By contrast, there was a statistically significant difference between teacher’s use of design (U= 6,179.4, r=0.138, p=0.046) for generalists (n=101) and specialists (n=106), with small effect size. Specialists reported a higher use of design than generalists, as shown in Figure 3.

**3.4.3 Use of design & use of planning (paired).** Comparing the reported use of design and use of planning for each teacher, there was a significant preference (higher ranking) for the use of planning over design (Z=-6.628, r=-0.6595, p<0.00), with a large effect size. 21% tied their ranking (n=21), 68% ranked planning higher than design (n=69) and 11% vice versa (n=11).

**3.4.4 Use of design & length of time teaching programming.** Over time, the most popular response for teachers was ‘Sometimes’



**Figure 4: Bar chart of the reported use of design by experience teaching programming for specialist teachers**



**Figure 5: Bar chart of the reported use of design by experience teaching programming for generalist teachers**

**Table 10: Cross tabulation of usefulness & use of design**

Useful-ness/Use	Always	Usually	Some-times	Rarely	Never	Totals
Essential	34%(37)	25%(27)	27%(30)	8%(9)	6%(7)	53%(110)
Very	7%(4)	30%(8)	46%(27)	12%(7)	5%(3)	28%(59)
Somewhat	3%(1)	9%(3)	53%(17)	19%(6)	16%(5)	15%(32)
A little	0%(0)	17%(1)	17%(1)	66%(4)	0%(0)	3%(6)
Totals	20%(42)	24%(49)	36%(75)	13%(26)	7%(15)	207

(36.2%), but broadly the longer teachers had been teaching programming, the more likely they were to report a higher use of design (n=207,  $\chi^2=20.353$ , d.f.=4, p<.000) with moderate effect size. Comparing generalists & specialists teachers in more detail, specialists were more likely to use design as their number of years’ experience increases (see Figure 4). This was similar for generalists, but with ‘Never’ and ‘Rarely’ more prominent in 1 and 3 years’ experience bands. ‘Sometimes’ featured in 2 to 3 years and dominated from year 3 on (see Figure 5).

**Table 11: Paired comparison of usefulness & use of design**

Group	Tied ranking	Use ranks higher	Usefulness ranks higher
All	37% (76)	5% (10)	58% (121)
Generalists	39% (39)	5% (5)	56% (57)
Specialists	35% (37)	5% (5)	60% (64)

**Table 12: Cross tabulation of usefulness & use of planning**

Usefulness/Use	Always	Usually	Some -times	Rarely	Never	Totals
Essential	<b>75%(46)</b>	<b>20%(12)</b>	3%(2)	2%(1)	0%(0)	60%(61)
Very	<b>25%(6)</b>	<b>58%(14)</b>	13% (3)	0%(0)	4%(1)	24%(24)
Somewhat	<b>25%(3)</b>	<b>42%(5)</b>	<b>25%(3)</b>	8%(1)	0%(0)	12(12%)
A little	50%(1)	0%(0)	50% (1)	0%(0)	0%(1)	2%(2)
Not at all	100%(2)	0%(0)	0% (0)	0%(0)	0%(0)	2%(2)
Totals	57%(58)	31%(31)	9%(9)	2%(2)	1%(1)	101

### 3.5 Usefulness & use of planning & design

We first report independent statistics, then paired statistics where *Usefulness* and *use* responses are paired across their scales from 'Essential' = 'Always' used to 'Not at all useful' = 'Never' used.

**3.5.1 Usefulness of design & use of design.** There was a statistically significant difference between the usefulness and use of design ( $\chi^2=32.281$ ,  $d.f=4$ ,  $p<0.000$ ). Despite most teachers (82%) rating usefulness of design as 'Essential' or 'Very Useful' they did not carry this through to a similar level of use. Use of design was relatively evenly spread across the ratings 'Always' (20%), 'Usually' (24%), 'Sometimes' (36%) and 'Rarely' or 'Never' (20%). There was a broad correlation of the higher usefulness ratings with the use ratings & the lower usefulness and use ratings (shown **bold** in Table 10).

Teachers' responses related to use and usefulness of design were paired across the scales and a statistically significant difference seen ( $Z=-9.084$ ,  $r=-0.631$ ,  $p<0.000$ ). Over a third used design in line with reported usefulness, but, 58% used design less. The same test was applied for both generalists ( $n=101$ ) ( $Z=-6.069$ ,  $r=.603$ ,  $p<0.000$ ) and specialists ( $n=104$ ) ( $Z=-6.733$ ,  $r=0.66$ ,  $p<0.000$ ) and found to be significant for both with large effect size (see Table 11).

**3.5.2 Usefulness of planning & use of planning.** By contrast, teachers' planning use was high, as was reported usefulness of planning ( $n=101$ ,  $\chi^2=18.431$ ,  $d.f=4$ ,  $p=0.001$ ): see Table 12. Comparing paired responses for generalist teachers' ( $n=101$ ) reported usefulness and use of planning, **no** significant difference was found ( $Z=-.99$ ,  $r=.09$ ,  $p=0.921$ ). 62.3% ( $n=63$ ) teachers tied their use and usefulness of planning ratings, 19.8% ( $n=20$ ) reported a higher use than usefulness and 19.8% ( $n=20$ ) vice versa. Reported use of planning was in line with its reported usefulness.

### 3.6 Detailed design & planning questions

Descriptive statistics for the paired design and planning questions are shown in Table 14 & 13. The paired statistic (Wilcoxon Signed Rank) with rank changes is shown in Table 15. For each paired question we also provide a cross tabulation showing % of responses, number of responses (n) and **accumulative** % of responses.

**Table 13: Descriptive data for detailed design questions**

Question	n	Mode	Mean
Demonstrates	186	'Usually'	2.36
Annotates	187	'Sometimes'	3.19
Refers to	188	'Usually'	2.29
Differentiates	187	'Sometimes'	3.36
Creates first	190	'Sometimes'	2.56

**Table 14: Descriptive data for detailed planning questions**

Question	n	Mode	Mean
Demonstrates	100	'Always'	1.61
Annotates	100	'Always'	1.76
Refers to	100	'Always'	1.3
Differentiates	96	'Usually'	2.19
Creates first	100	'Always'	1.5

**Table 15: Paired comparison of detailed design & planning questions for generalists**

Question	Tied ranking	Plan ranks higher	Design ranks higher	Test Statistics
Demonstrates	33%(29)	54%(47)	13%(11)	$n=87$ , $Z=-4.947$ , $r=-0.530$ , $p<0.00$
Annotates	18%(16)	73%(63)	9%(8)	$n=87$ , $Z=-6.440$ , $r=-0.690$ , $p<0.00$
Refers to	37%(33)	56%(49)	7%(6)	$n=88$ , $Z=-5.849$ , $r=0.623$ , $p<0.00$
Differentiates	23%(20)	70%(61)	7%(6)	$n=87$ , $Z=-6.613$ , $r=-0.709$ , $p<0.00$
Creates first	23%(21)	71%(63)	6%(5)	$n=89$ , $Z=-6.617$ , $r=-0.701$ , $p<0.00$

There were no statistically significant differences between the groups of male and female & generalists and specialists in any of the five paired questions. This data is available on request.

**3.6.1 Demonstrating (modelling) the creation of designs/plans.** Table 16 shows responses for the questions 'For programming projects, do you demonstrate to pupils (model) the creation of designs?' & 'For substantial writing tasks, do you demonstrate to pupils (model) the creation of plans?' Of note, 30% 'Always' modelled design compared to 55% for planning.

**3.6.2 Annotate design /annotate plan.** Table 17 shows responses for the questions 'Based on your recent practice of teaching programming projects, did you expect pupils to annotate their design with useful code snippets?' & 'Based on your recent practice of teaching substantial writing tasks, did you expect pupils to annotate their plan with useful words & phrases?' Of note, 42% 'Rarely' or 'Never' annotated design compared to only 3% for planning.

**3.6.3 Refer to design /refer to plan.** Table 18 shows responses for questions 'Based on your recent practice of teaching programming projects, did you expect pupils to refer to their design when coding'

**Table 16: Teachers demonstrate (model) creation of designs or plans**

Use	Design			Planning		
	%	n	Accum%	%	n	Accum%
Always	30%	55	<b>30%</b>	55%	55	<b>55%</b>
Usually	33%	62	<b>63%</b>	31%	31	<b>86%</b>
Sometimes	19%	36	<b>82%</b>	12%	12	<b>98%</b>
Rarely	7%	13	<b>89%</b>	2%	2	<b>100%</b>
Never	11%	20	<b>100%</b>	0%	0	<b>100%</b>

**Table 17: Teachers expect pupils to annotate designs or plans with useful commands or vocabulary**

Use	Design			Planning		
	%	n	Accum%	%	n	Accum%
Always	12%	22	<b>12%</b>	50%	50	<b>50%</b>
Usually	13%	24	<b>25%</b>	28%	28	<b>78%</b>
Sometimes	33%	63	<b>58%</b>	19%	19	<b>97%</b>
Rarely	28%	52	<b>86%</b>	2%	2	<b>99%</b>
Never	14%	26	<b>100%</b>	1%	1	<b>100%</b>

**Table 18: Teachers expect pupils to refer to designs or plans during programming or writing**

Use	Design			Planning		
	%	n	Accum%	%	n	Accum%
Always	30%	57	<b>30%</b>	76%	76	<b>76%</b>
Usually	32%	60	<b>62%</b>	17%	17	<b>93%</b>
Sometimes	20%	37	<b>82%</b>	7%	7	<b>100%</b>
Rarely	14%	27	<b>96%</b>	0%	0	<b>100%</b>
Never	4%	7	<b>100%</b>	0%	0	<b>100%</b>

**Table 19: Teachers use designs or plans to differentiate work**

Use	Design			Planning		
	%	n	Accum%	%	n	Accum%
Always	5%	9	<b>5%</b>	28%	27	<b>28%</b>
Usually	20%	38	<b>25%</b>	33%	33	<b>61%</b>
Sometimes	33%	61	<b>58%</b>	29%	28	<b>90%</b>
Rarely	18%	34	<b>76%</b>	10%	9	<b>100%</b>
Never	24%	45	<b>100%</b>	0%	0	<b>100%</b>

& 'Based on your recent practice of teaching substantial writing tasks, did you expect pupils to refer to their plan during writing?' Of note, 30% 'Always' expected pupils to refer to design during coding compared to 76% for planning.

*3.6.4 Use designs/plans to differentiate work.* Table 19 shows responses for the questions 'For programming projects, do you use designs to differentiate work?' & 'For substantial writing tasks, do you use plans to differentiate work?' Of note, 24% 'Never' used design to differentiate but no teachers did the same for planning.

**Table 20: Teachers expect pupils to create designs or plans before coding or writing starts**

Use	Design			Planning		
	%	n	Accum%	%	n	Accum%
Always	15%	28	<b>15%</b>	62%	62	<b>62%</b>
Usually	32%	62	<b>47%</b>	28%	28	<b>90%</b>
Sometimes	37%	70	<b>84%</b>	9%	9	<b>99%</b>
Rarely	13%	25	<b>97%</b>	0%	0	<b>99%</b>
Never	3%	5	<b>100%</b>	1%	1	<b>100%</b>

*3.6.5 Create design /plan before coding/writing.* Table 20 shows responses for the questions 'Based on your recent practice of teaching programming projects, did you expect pupils to create a design before they start coding?' & 'Based on your recent practice of teaching substantial writing tasks, did you expect pupils to create a plan before they start to write?' Of note, 15% 'Always' expected a design to be created first compared to 62% for planning.

## 4 DISCUSSION

We first consider possible biases then organise our discussion around confidence, use and usefulness and specific uses.

### 4.1 Participants & Survey: possible biases

Our population had a bias of more experienced teachers of writing than programming. Almost three times had more than 5 years' experience in writing (64%) versus programming (23%) (see Figures 2 and 1). More experienced writing teachers may be more familiar with planning than programming teachers with design.

However, participants' interpretation of the number of years they have been teaching programming is interesting. Before the 2014 English Computing Curriculum was mandated, teachers were required to teach 'control' in the Information and Communication Technology curriculum, including programming of physical devices [6]. Therefore, a generalist teacher with 5 or more years experience might be expected to have 5 or more years of teaching experience related to 'control'. Why teachers did not classify such experience as programming merits further investigation.

We were surprised by the high % of specialist teachers (51%) who responded to our survey. There may be a trend of specialists to deliver computing in primary schools. On the other hand, it may be that our advertising was not accessible to generalists and therefore there is a potential bias in our population. However, this high % of specialists has provided us with an opportunity to investigate this group.

### 4.2 RQ1: Are teachers' confidence to teach programming & writing the same?

We asked simple questions about confidence to set the scene and to compare with our other survey questions. We found our teachers' confidence to teach programming and writing were different.

Gender self-efficacy of academic subjects is a complex area, but some have suggested that females may have higher confidence in art and literacy compared to science and mathematics [13]. In line with this proposal our females reported higher confidence to teach writing than males (Table 4), with small effect size. In

contrast, males reported higher confidence to teach programming than writing (Table 3), with a moderate effect size. In opposition to this 'preference', there was no statistically significant difference between female and male **specialist** teachers reported confidence to teach programming ( $n=104$ ,  $U=1003$ ,  $r=-0.1893$ ,  $p=0.54$ ) with small effect size. K-5 female specialist teachers may be an avenue of fruitful investigation to discover contributing factors to this group's pattern of increased confidence compared to generalist females.

Unsurprisingly, specialist teachers reported a higher degree of confidence than generalists (see Table 5) to teach programming. Similarly the longer teachers had been teaching programming, the more confident they were to deliver programming lessons (see Figure 1). However, the length of time it took teachers to become predominantly confident is of particular interest. Per Figure 1, until teachers had taught programming for 'more than 5 years', '1 Very Confident' ranked lower than other ratings. This has implications for expectations as workforces are supported to deliver computing.

Despite confidence being statistically significantly different for males and females for teaching writing and programming, no further statistically significant differences were found for gender comparisons. How self-efficacy translates into action taken is a complex area [2] and we will continue to look for patterns in future work reviewing other survey questions.

### 4.3 RQ2: Are there relationships between usefulness & use of design & planning?

Research question RQ2, is subdivided into three questions, RQ2-1 to RQ2-3, comparing usefulness of design and planning, use of design and planning and usefulness and use, we address each in turn.

**RQ2-1: Are teachers' perceptions of the usefulness of design in programming & usefulness of planning in writing the same?** Usefulness of both planning and design were highly rated by teachers. 82% of teachers ( $n=207$ ) reported that design was 'Essential' or 'Very Useful' in teaching programming, with 78% for planning in teaching writing (Table 6). However, a closer inspection of a paired comparison of teachers' ratings of usefulness, revealed that specialists regarded design as more useful, compared to generalists who ranked planning as more useful (Table 8). If specialist teachers are becoming more prevalent in K-5 English schools, then such underlying preferences may become more significant and merit further investigation.

**RQ2-2: Are teachers' reported use of design in programming & use of planning in teaching writing the same?** Teacher's use of design in programming and use of planning in teaching writing was not the same. 88% of our teachers 'Always' or 'Usually' required their pupils to use a plan at some point during writing, whereas half of this, only 44% of teachers, 'Always' or 'Usually' required pupils to consider or use design during programming. The most popular use for design dropped to 'Sometimes' for most teachers (33%), as shown in Table 9. Specialists were more likely to use design than generalist teachers (Figure 3). This use increased with years teaching programming (Figures 4 & 5). However, the different patterns of use, over time, were very different, with much lower use by generalists in the early part of their programming teaching career. We will be correlating this to our other data, such as training received, confidence, planning & resources used in next work.

**RQ2-3: Are there relationships between usefulness and use of design & planning?** 88% of the generalist teachers 'Always' or 'Usually' used planning in teaching writing in-line with 78% viewing planning as being 'Essential' and 'Very Useful' (Table 12). The picture was different for design. The reported usefulness of design (82%) and its use (44%) were not in-line. Teachers who taught programming did not follow through on their use of design compared to their reported usefulness of it (Table 10).

Specialist teachers did follow through to a higher degree than generalists. Specialists valued (Table 8) and used design (Figure 3) more than their generalist counterparts. Similarly, as teachers became more experienced, the use of design increased (Figures 4 & 5). During the 1-3 year experience period a high proportion of generalists reported 'Never' or 'Rarely' using design and had low confidence, but at the same time perceived design as being useful, and were using planning in teaching writing. Simply because instructional techniques are perceived as useful and are used in one subject and perceived as useful in a second subject may not result in a seamless transfer. The transfer of knowledge, skill and understanding, including instructional techniques, from one context to another is complex [22] as are the choices made by teachers as to what instructional strategy or method is most effective. Professional development received, curriculum requirements, experience of strategies used, self-efficacy and students needs and performance have been cited as drivers for such choices [21]. Our survey asked about teachers' professional development as well as their uses of curriculum products. We are analysing this data in ongoing work.

### 4.4 RQ3: Do teachers use design and planning in the same way for specific uses?

Despite teachers, across the five detailed questions, not using the techniques as frequently in programming as they did in writing, they *were* still using the techniques with the most popular uses cited as 'Sometimes' or 'Usually' (Table 13).

As generalists and specialists had reported a significant difference in their overall use of design (Figure 3), we expected this to be mirrored in the detailed uses. It was not. There were no statistically significant differences for these groups. Similarly, we explored differences between male and female uses of instructional approaches to contrast to their reported differences in confidences of teaching programming and writing. Again this was not seen.

Using the mean responses to the questions on specific uses of planning (see Table 14) there were similarly high reports of all uses. With all average reports being between 'Always' and 'Usually', apart from using planning to differentiate work which had an average response of between 'Usually' and 'Sometimes'.

The picture was very different for design. Average responses sat between 'Usually' and 'Sometimes' for: *refers to*, *demonstrates* and *creates first*. They sat between 'Sometimes' and 'Rarely' for *annotates* and *differentiates*. Differentiating with planning and design was the least most popular use for both subjects.

Looking specifically at each of the specific uses RQ3 is discussed as separate research-subquestions, RQ3-1 to RQ3-5.

**RQ3-1: Do teachers demonstrate (model) the creation of designs & is this the same for planning?** Nearly two thirds (63%) of teachers 'Always' or 'Usually' modelled the creation of design

but nearly a fifth (18%) 'Rarely' or 'Never' did this. By comparison, 86% of teachers modelled the creation of plans in writing and only 2% 'Rarely' or 'Never' did this.

Modelling includes thinking aloud, a feature of discourse intensive pedagogy [11] on the choices being made, making mistakes to exemplify misconceptions and including pupils in decision making to involve them in their learning. Modelling has been identified as an essential teaching function in effective presentation of new material in the systematic instruction of teaching complex cognitive skills such as essay writing, reading comprehension and mathematical problem solving [26]. However, modelling requires teachers to reveal misconceptions and design choices available. Whether teachers are aware of this detail in the progression of teaching K-5 programming is not clear. Also, modelling may be a high stakes activity that requires teachers to be confident. How generalists with low confidence might deal with this challenge requires study.

**RQ3-2: Do teachers expect pupils to annotate designs with useful code snippets & is this the same for plans with useful words or phrases?** A quarter of teachers (25%) 'Always' or 'Usually' expected pupils to annotate plans with useful code snippets but 40% 'Rarely' or 'Never' expected this. By comparison, three-quarters of generalist teachers (78%) 'Always' or 'Usually' expected this and only 3% 'Rarely' or 'Never' required this for planning. In annotating designs with useful code snippets, pupils are bridging from 'design' to the 'code' [35]. The ability to recognise and then move between these 'levels of abstraction' has been cited as an important skill [1, 32]. Similarly, as they then use the design with its annotations, the task complexity is reduced, as they are reminded of the most important commands they need. There may be links here to the block model [28]: annotation may help pupils think about commands separately to the job of combining them into logical blocks and an overall structure. Annotated ideas recorded on the physical design artefact provide a historical snapshot of understanding to support assessment and development of a growth mindset as they can be reviewed and reflected on [35].

**RQ3-3: Do teachers expect pupils to refer to designs during coding & is this the same for planning during writing?** Nearly two-thirds of the teachers (62%) reported 'Always' or 'Usually' expecting pupils to refer to designs during coding and just less than a fifth (18%) 'Rarely' or 'Never'. Whereas, in teaching writing almost all teachers (93%) 'Always' or 'Usually' required their pupils to refer to plan while writing and no teachers reported 'Rarely' or 'Never' expecting this. Referring to the plan reduces cognitive load, as the plan serves as an 'aide memoire' of what has already been done, and what to do next. It also provides a 'personalized scaffolding map' that helps create a cohesive and complete solution and therefore increases the quality of their work' [35]. However, this supposes that gathering ideas first, or at some stage in a design, is the best approach for program development, see RQ3-5.

**RQ3-4: Do teachers use design to differentiate work & is this the same for planning?** A quarter of teachers (25%) 'Always' or 'Usually' used designs to differentiate work and 42% 'Never' or 'Rarely' expected this. This was a very different picture for planning where nearly two thirds (61%) 'Always' or 'Usually' used plans to differentiate and only 10% 'Rarely' or 'Never' used this instructional approach in writing. Planning can be used in a variety of

ways to differentiate writing work, the number of boxes of a storyboard or 'arms of a concept map' can be used to reduce content, coverage and complexity. The adding of annotated useful words and phrases can be more supported by teachers. Differentiation is a complex area. How it might be achieved and its effectiveness has been linked to teacher's understanding of pupil's current knowledge, skills and understanding through formative assessment [33]. Formative assessment relies on an understanding of learning intentions & success criteria [4]. In the teaching of writing, a high-stakes subject on which schools are measured nationally in England, such trajectories of 'low-level' goals are well known by teachers. However, in programming these are still in development [25]. Whether design is a useful component in this complicated teaching and learning process has not yet been evidenced, and may be impacted by whether design is implicit, explicit and created early or late in the development process.

**RQ3-5: Do teachers expect pupils to create a design before starting coding & is this the same for plans and writing?** Nearly half of the teachers (47%) expected pupils to 'Always' or 'Usually' create a design before programming and 15% said they 'Rarely' or 'Never' expected this. In writing, double the number of teachers (90%) reported 'Always' or 'Usually' expecting the same in writing and only 1% 'Rarely' or 'Never' expected this.

There are several different ways to start a programming project. Pupils could use, or copy, an existing solution and then modify it. In which case, design might be implicit as a thought process occurring during coding as choices are made. Or design might be explicit, exemplified as a design for the starter code which is expected to be modified in line with changes made. Choices made may be reactions to minimally guided exploration or following orchestrated teacher-led experiences. On the other hand, projects might start with a blank sheet, expecting pupils to develop solutions for tasks, again with or without design being involved. It has been suggested that learners might have a preference for whether they use a 'top-down' approach, starting with a design compared to exploring and building 'bottom-up' using a 'bricolage' approach [34]. However, novice learners, and their teachers, may not know what is doable [35] in which case a 'top-down' approach may be difficult and frustrating. There may be links here with our finding that design becomes more prevalent as teachers become more experienced, as they may start to know what is 'doable'. A use, modify, make [16] approach might, over time, combine all approaches, with or without design consideration. Or a combination of a bottom-up exploration and some top-down design might be adopted [29]. What impact early explicit design has on pupil outcomes and how this maps to teacher confidence and experience merits further investigation.

## 5 CONCLUSIONS

Our intention was to discover K-5 teachers' views on usefulness and use of design in programming and planning in writing. We found, in writing, our K-5 teachers reported planning as having high usefulness with 78% of them reporting it as 'Essential' or 'Very Useful'. This perception was translated into use with 88% reporting 'Always' or 'Usually' using planning. For specific instructional approaches, there were similarly high reports of use.

In programming, K-5 teachers reported a similar high usefulness of design. 82% said it was 'Essential' or 'Very useful', but this was

not converted to use to the same degree as in planning. Half as many teachers as for planning (44%) reported that they used design 'Always' or 'Usually'. Similarly, the use of specific instructional approaches associated with design was lower than for planning.

However, some K-5 teachers 'Always' use design in the same way that teachers use planning in teaching writing. Specialists were more likely to use design, & generalists more likely to use planning. However, these differences did not carry forward to detailed use. Similarly, we found that the longer one taught programming, the more likely one was to report using design. We discussed what is contributing to the variance in use, including 'bottom-up' versus 'top-down approaches' to running projects, K-5 teachers' knowledge of progression in programming & the complexity of transferring pedagogical knowledge from one subject to another.

We also found female K-5 teachers reported higher confidence to teach writing than males and vice versa for programming. However, specialist females reported no statistically significant difference in confidence to teach programming compared with their K-5 male colleagues. Further research is needed to investigate the popularity, views and experiences of specialist teachers.

In teaching K-5 writing, planning has been recognised as a high value, high impact instructional approach for supporting pupil progress. It is also a familiar strategy for K-5 generalist teachers. Whether these benefits & this teacher expertise can be transferred from writing to K-5 programming merits further investigation, not only for its potential impact on pupil progress but also for impact on teacher confidence.

In future work, our analysis will explore in more detail all survey questions, their synergies & combine this with previous teacher interviews, pupil interviews & next step focus groups. There is much to be done investigating the synergies between successful teaching strategies used in other subjects and the teaching of programming.

## REFERENCES

- [1] Michal Armoni. 2013. On Teaching Abstraction in Computer Science to Novices. *Journal of Computers in Mathematics and Science Teaching* 32, 3 (2013), 265–284.
- [2] Albert Bandura and VS Ramachaudran. 1994. Self-efficacy. Encyclopedia of human behavior. VS Ramachaudran (Ed.) (1994).
- [3] M.U. Bers. 2017. *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*. Taylor & Francis. <https://books.google.co.uk/books?id=iWEwDwAAQBAJ>
- [4] Paul Black and Dylan Wiliam. 2009. Developing the theory of formative assessment. *Educational Assessment, Evaluation and Accountability (formerly: Journal of Personnel Evaluation in Education)* 21, 1 (2009), 5.
- [5] Louis Cohen, Lawrence Manion, and Keith Morrison. 2011. *Research methods in education*. Vol. 7th Edition. Routledge.
- [6] DfE. 1999. The national curriculum for England ICT. Online. (1999). <http://webarchive.nationalarchives.gov.uk/20100202100434/http://curriculum.qcda.gov.uk/key-stages-1-and-2/subjects/ict/keystage1/index.aspx>
- [7] DfE. 2013. Computing programmes of study key stages 1 and 2 National Curriculum in England. (Sept 2013). <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study> May 5, 2015.
- [8] Katrina Falkner and Rebecca Vivian. 2015. A review of computer science resources for learning and teaching with K-12 computing curricula: An Australian case study. *Computer Science Education* 25, 4 (2015), 390–429.
- [9] Steve Graham, Alisha Bollinger, C Olson, Catherine DAoust, Charles MacArthur, Deborah McCutchen, and Natalie Olinghouse. 2012. Teaching elementary school students to be effective writers. *What Works Clearinghouse, US Department of Education* (2012).
- [10] Grover, Pea, and Cooper. 2015. Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education* 25, 2 (2015), 199–237.
- [11] Shuchi Grover and Roy Pea. 2013. Using a discourse-intensive pedagogy and android's app inventor for introducing computational concepts to middle school students. In *Proceeding of the 44th ACM technical symposium on Computer science education*. ACM, 723–728. <https://doi.org/10.1145/2445196.2445404>
- [12] Steve Higgins, Dimitra Kokotsaki, and Robert Coe. 2012. The teaching and learning toolkit. *Education Endowment Foundation and Sutton Trust* (2012).
- [13] Chiungjung Huang. 2013. Gender differences in academic self-efficacy: a meta-analysis. *European journal of psychology of education* 28, 1 (2013), 1–35.
- [14] Yasmin B Kafai and Veena Vasudevan. 2015. Constructionist Gaming Beyond the Screen: Middle School Students' Crafting and Computing of Touchpads, Board Games, and Controllers. In *Proceedings of the Workshop in Primary and Secondary Computing Education*. ACM, 49–54.
- [15] Petra Kastl and Ralf Romeike. 2015. Now they just start working, and organize themselves First Results of Introducing Agile Practices in Lessons. In *Proceedings of the Workshop in Primary and Secondary Computing Education*. ACM, 25–28.
- [16] Irene Lee, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith, and Linda Werner. 2011. Computational thinking for youth in practice. *ACM Inroads* 2, 1 (2011), 32–37. <https://doi.org/10.1145/1929887.1929902>
- [17] Dastyni Loksa, Andrew J Ko, Will Jernigan, Alannah Oleson, Christopher J Mendez, and Margaret M Burnett. 2016. Programming, problem solving, and self-awareness: effects of explicit guidance. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1449–1461.
- [18] Elizabeth Mavhunga, Bashirah Ibrahim, Makomosela Qhobela, and Marissa Rollnick. 2016. Student Teachers Competence to Transfer Strategies for Developing PCK for Electric Circuits to Another Physical Sciences Topic. *African Journal of Research in Mathematics, Science and Technology Education* 20, 3 (2016), 299–313.
- [19] Gary E Meek, Ceyhun Ozgur, and Kenneth Dunning. 2007. Comparison of the t vs. Wilcoxon signed-rank test for Likert scale data and small samples. *Journal of Modern Applied Statistical Methods* 6, 1 (2007), 10.
- [20] Philip Morgan and Sid Bourke. 2008. Non-specialist teachers' confidence to teach PE: the nature and influence of personal school experiences in PE. *Physical Education and Sport Pedagogy* 13, 1 (2008), 1–29.
- [21] William D Nichols, Luana J Zellner, Victor L Willson, Sandra Mergen, and Carl A Young. 2005. What affects instructional choice? Profiles of K-2 teachers' use of reading instructional strategies and methods. *Journal of Literacy Research* 37, 4 (2005), 437–458.
- [22] Roy D Pea. 1987. Socializing the knowledge transfer problem. *International Journal of Educational Research* 11, 6 (1987), 639–663.
- [23] Jacob Perrenet, Jan Friso Groote, and Eric Kaasenbrood. 2005. Exploring students' understanding of the concept of algorithm: levels of abstraction. *ACM SIGCSE Bulletin* 37, 3 (2005), 64–68. <https://doi.org/10.1145/1067445.1067467>
- [24] Kathryn Rich, Carla Strickland, and Diana Franklin. 2017. A Literature Review through the Lens of Computer Science Learning Goals Theorized and Explored in Research. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. ACM, 495–500.
- [25] Kathryn M Rich, Carla Strickland, T Andrew Binkowski, Cheryl Moran, and Diana Franklin. 2017. K-8 Learning Trajectories Derived from Research Literature: Sequence, Repetition, Conditionals. In *Proceedings of the 2017 ACM Conference on International Computing Education Research*. ACM, 182–190.
- [26] Barak V Rosenshine. 1986. Synthesis of research on explicit teaching. *Educational leadership* 43, 7 (1986), 60–69.
- [27] Tanya Santangelo and Natalie G Olinghouse. 2009. Effective writing instruction for students who have writing difficulties. *Focus on exceptional children* 42, 4 (2009), 1.
- [28] Carsten Schulte. 2008. Block Model: an educational model of program comprehension as a tool for a scholarly approach to teaching. In *Proceedings of the Fourth international Workshop on Computing Education Research*. ACM, 149–160.
- [29] Carsten Schulte, Johannes Magenheimer, Kathrin Müller, and Lea Budde. 2017. The design and exploration cycle as research and development framework in computing education. In *Global Engineering Education Conference (EDUCON), 2017 IEEE*. IEEE, 867–876. <https://doi.org/10.1109/EDUCON.2017.7942950>
- [30] Lee S Shulman. 1986. Those who understand: Knowledge growth in teaching. *Educational researcher* 15, 2 (1986), 4–14.
- [31] Royal Society. 2017. After the reboot: computing education in UK schools. Online, The Royal Society, 6 ÅŠ 9 Carlton House Terrace, London SW1Y 5AG. [royalsocietypublishing.org/computing-education](https://royalsocietypublishing.org/computing-education)
- [32] David Statter and Michal Armoni. 2016. Teaching Abstract Thinking in Introduction to Computer Science for 7th Graders. In *Proceedings of the 11th Workshop in Primary and Secondary Computing Education*. ACM, 80–83. <https://doi.org/10.1145/2978249.2978261>
- [33] Carol A Tomlinson and Tonya R Moon. 2013. *Assessment and student success in a differentiated classroom*. ASCD.
- [34] Sherry Turkle and Seymour Papert. 1992. Epistemological pluralism and the revaluation of the concrete. *Journal of Mathematical Behavior* 11, 1 (1992), 3–33.
- [35] Jane Waite, Paul Curzon, D Marsh, Sue Sentance, and A Hawden-Bennett. 2018. Abstraction in action: K-5 teachers' uses of levels of abstraction, particularly the design level, in teaching programming. *International Journal Of Computer Science Education In Schools* (2018).
- [36] Jane Waite, Paul Curzon, William Marsh, and Sue Sentance. 2016. Abstraction and common classroom activities. In *Proceedings of the 11th Workshop in Primary and Secondary Computing Education*. ACM, 112–113.