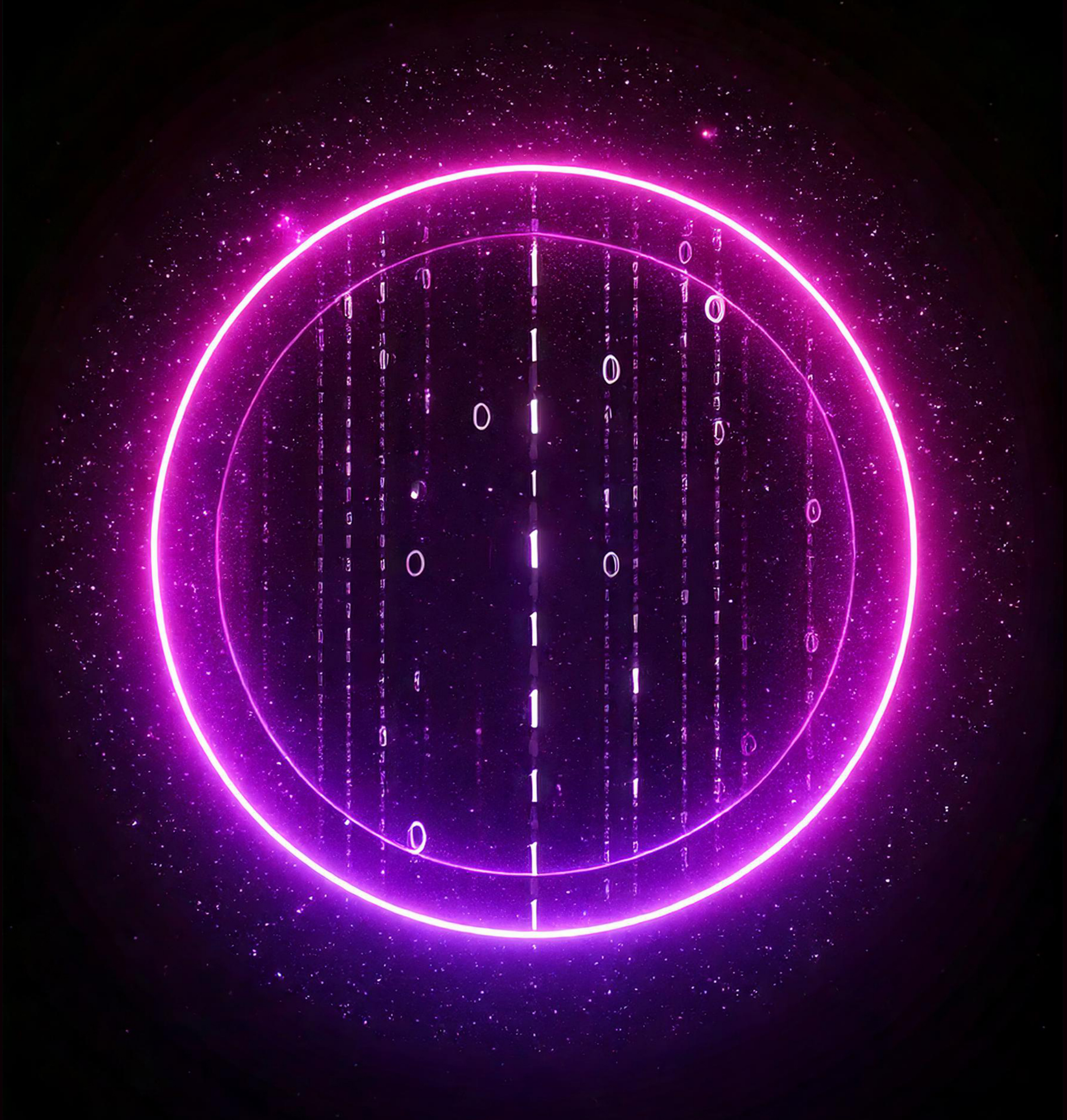
PROGRAMMING IN SECONDARY EDUCATION IN ENGLAND

TECHNICAL REPORT



Alex Hadwen-Bennett
Peter EJ Kemp

October 2024

Centre for Research in Education in STEM

KING'S College LONDON

Programming in Secondary Education in England

Technical Report

Alex Hadwen-Bennett
Peter EJ Kemp

October 2024

Abstract

This report provides an analysis of a national survey of 108 Computing teachers between January and April 2024. The research was undertaken by the Computing Education Research team at King's College London to examine current practices in teaching programming education in England.

Key findings:

- Python dominates the programming landscape in schools with 90%+ of KS3 and KS4 teachers using the language. KS5 provision offers more variety in languages with C# being present in 30% of schools.
- State schools are more likely to have programming taking place in 0-25% of KS3 and KS4 lessons, whereas independent schools are more likely to have programming taking place in 26-50% and of KS3 lessons.
- PRIMM is the most popular programming pedagogical technique, with female teachers being more likely to use this technique "often" than males. Male teachers are more likely to often use Parson's problems than female teachers.
- The majority of teachers have taught students better at programming than themselves, with female teachers more likely to say this than male teachers. In mixed schools, the gender split of best programmers is roughly in line with national figures for entry to GCSE computer science.
- Most schools offer a computing-related club, but state non-selective schools are the least likely to. Independent schools are more likely to offer Robotics clubs.
- Competitions are popular, with 74% of all schools supporting students in entering competitions, including 100% of the independent schools surveyed supporting entries. Independent schools are more likely to enter students into programming competitions, whereas state-maintained schools are more likely to enter students into game design and animation competitions.

Contents

Background and introduction

Since the introduction of the new national curriculum in 2014, all children in England have been required to learn how to program a computer (DfE 2014). There has been some success in introducing computer science into schools across the country (Kemp et al. 2024a) with “computer and code clubs” being the most common clubs of any type offered in primary and secondary schools (IFF Research 2023). However, learning to program a computer is considered to be a challenging endeavour for most learners due to the multiple levels of abstraction involved (Blackwell 2002). A considerable effort has gone into developing tools and strategies to support learners in the process, particularly those who struggle to access programming (e.g. NCCE / PRIMM / Teaspoon). The survey outlined in this report was designed to provide background contextual information regarding the current practices employed when teaching programming in secondary schools in England. It forms a part of a larger project which seeks to explore the learning journeys of highly able programmers.

Methodology

We designed an online survey which invited teachers who deliver GCSE[1] Computer Science in England to answer questions covering the following areas:

- Demographic information and programming background,
- Clubs, Extra-curricular activities and competitions,
- Programming languages taught at KS3/4[2],
- Programming pedagogies employed,
- Perceptions of highly able programmers.

The survey was designed to be anonymous, however participants were asked to either provide information regarding the characteristics of their school/college or its

[1] General Certificate of Secondary Education – the standard qualification taken by 15/16 year olds in England, Wales and Northern Ireland

[2] The stages of education in England are split into 5 key stages. Key stages 3, 4 and 5 apply to high school. KS3: ages 11-14 KS4: ages 15-16 (when students might take GCSEs) KS5: ages 17-18 (when students might take A-Levels)

Postcode/URN so this information could be obtained to enable statistical comparisons between responses from different types of provider.

Ethical clearance for the study was sought and obtained from the King's College London Ethics Committee (*Ethical Clearance Reference Number:* **LRS/DP-23/24-37000**).

Distribution

The survey was distributed between January and April 2024 through organisations involved in Computing education in England such as CAS (Computing at School) and examination boards. Additionally, it was also shared through social networks such as Twitter/X, LinkedIn and closed Facebook groups for Computing teachers.

There were 166 responses to the survey, however, only 108 of these were complete responses. The results below analyse the full responses.

Analysis

In this initial report on the findings from the survey, descriptive statistics were built from the responses given by participants. In several instances below results are analysed against the gender of the teacher, or the type of school that the respondent teaches in.

Contextual information

Although we received responses from across England, Table 1 demonstrates that a large proportion of the responses (58.4%) originated from London and the South of England. This is likely to be related to the research team and much of their professional network being based in London.

Table 1: School/College Location

Region	n	%
East Midlands	4	3.7
East of England	7	6.5
London	33	30.6
North East	4	3.7
North West	6	5.6
South East	18	16.7
South West	12	11.1
West Midlands	14	13.0
Yorkshire and the Humber	4	3.7
NA	6	5.6

Table 2 shows that over half (69.4%) of the responses came from state-maintained schools (Comprehensive / Secondary Modern / Free Schools / University technology colleges), 8.3% from grammar schools and 16.7% from independent schools. Additionally, as can be seen in Table 3, over three quarters of the responses come from schools with a mixed gender intake. Nationally, 87.7% of KS4 students attend a comprehensive school, 4.1% attend a grammar and 8.3% attend a private school. Splitting population by gender, 6.9% attended all girls schools, 4.5 % all boys and 88.7% mixed[3]. Therefore, the responses to the survey show an over representation of grammar and independent schools in the survey results, as well as an over representation of single sex schools, potentially a reflection of increased tendency amongst grammar and private schools to be single sex schools.

Table 4 shows that the teachers who responded to the survey were nearly evenly split between the genders, 45.4% female and 53.7% male. This contrasts with the national statistics which show that 41% of teachers were female (Kemp et al. 2024b).

[3] national statistics calculated from the 2023 Department for Education KS4 results tables, excluding special schools.

Table 2: School/College Type

Type	n	%
Comprehensive / Secondary Modern / Free	75	69.4
Grammar	9	8.3
Private / independent	18	16.7
NA	6	5.6

Table 3: School/College Gender

Gender	n	%
Boys	6	5.6
Girls	12	11.1
Mixed	84	77.8
NA	6	5.6

Table 4: Teacher gender

Gender	n	%
Female	49	45.4
Male	58	53.7
Prefer not to say	1	0.9

Programming within the secondary curriculum

We begin by exploring the results which provide an insight into the ways in which programming is currently being taught in secondary schools in England. This will include a discussion of the programming languages being employed in the classroom, the amount lesson time dedicated to programming, the pedagogical approaches employed by teachers and the perceived programming expertise of teachers in relation to the best programmers they have taught.

Programming languages taught

We will examine the programming languages being used to teach programming in secondary schools and colleges. Table 5 provides an overview of the languages that are taught at different key stages. It is important to note that respondents were able to select multiple languages.

Python has been a popular programming language for education in England for some time as indicated in the Royal Society (2017) 'After the Reboot' report in which Python topped the list of programming languages used in secondary education. Our results demonstrate that this continues to be the case, with over 90% of schools using Python at KS3 and KS4. At KS5 this dips to 75% with languages such as C#, Java and Javascript increasing. The use of Javascript at KS5 shows the biggest increase between key stages, going from 9.3% at KS4 to over 52.5%. This increase is likely to be partially explained by the number of A-Level students choosing to develop web applications for their NEA (Non-Examined Assessment)[4]. Scratch and other block-based languages continue to be a popular choice at KS3, with over 60% of schools using it with this age group. The use of block-based languages drops to less than 3% at KS4 and 5. However, this is likely to be related to them not currently being accepted for assessments in the main GCSE and A-Level Computer Science examination specifications in England.

[4] The Non-Examined Assessments in Computer Science take the form of what is traditionally thought of as coursework

Table 5: Programming languages taught

		Python		Scratch		VB		C#		JS		Java		Pascal	
KS	**Total**	**n**	**%**	**n**	**%**	**n**	**%**	**n**	**%**	**n**	**%**	**n**	**%**	**n**	**%**
Clubs	108	41	38.0	19	17.6	-	-	5	4.6	-	-	-	-	-	-
Don't	108	3	2.8	36	33.3	100	92.6	79	73.1	58	53.7	97	89.8	107	99.1
KS3	103	95	92.2	66	64.1	2	1.9	2	1.9	14	13.6	2	1.9	-	-
KS4	108	101	93.5	3	2.8	4	3.7	5	4.6	10	9.3	1	0.9	-	-
KS5	80	60	75.0	1	1.2	5	6.2	24	30.0	42	52.5	7	8.8	1	1.2

Lesson time dedicated to programming

The participants were asked the percentage of lesson time that they spend on programming at different key stages. Figure 1 shows that at KS3 42.7% of schools spend 0-25% of lesson time on programming with 38.8% spending 25-50%. At KS4, almost half (49.1%) of schools and colleges spend 26-50% of lesson time programming, whereas 37% spend 51-75%. A similar pattern is observed at KS5 with 45% spending 26-50% and 40% spending 51-75% of lesson time.

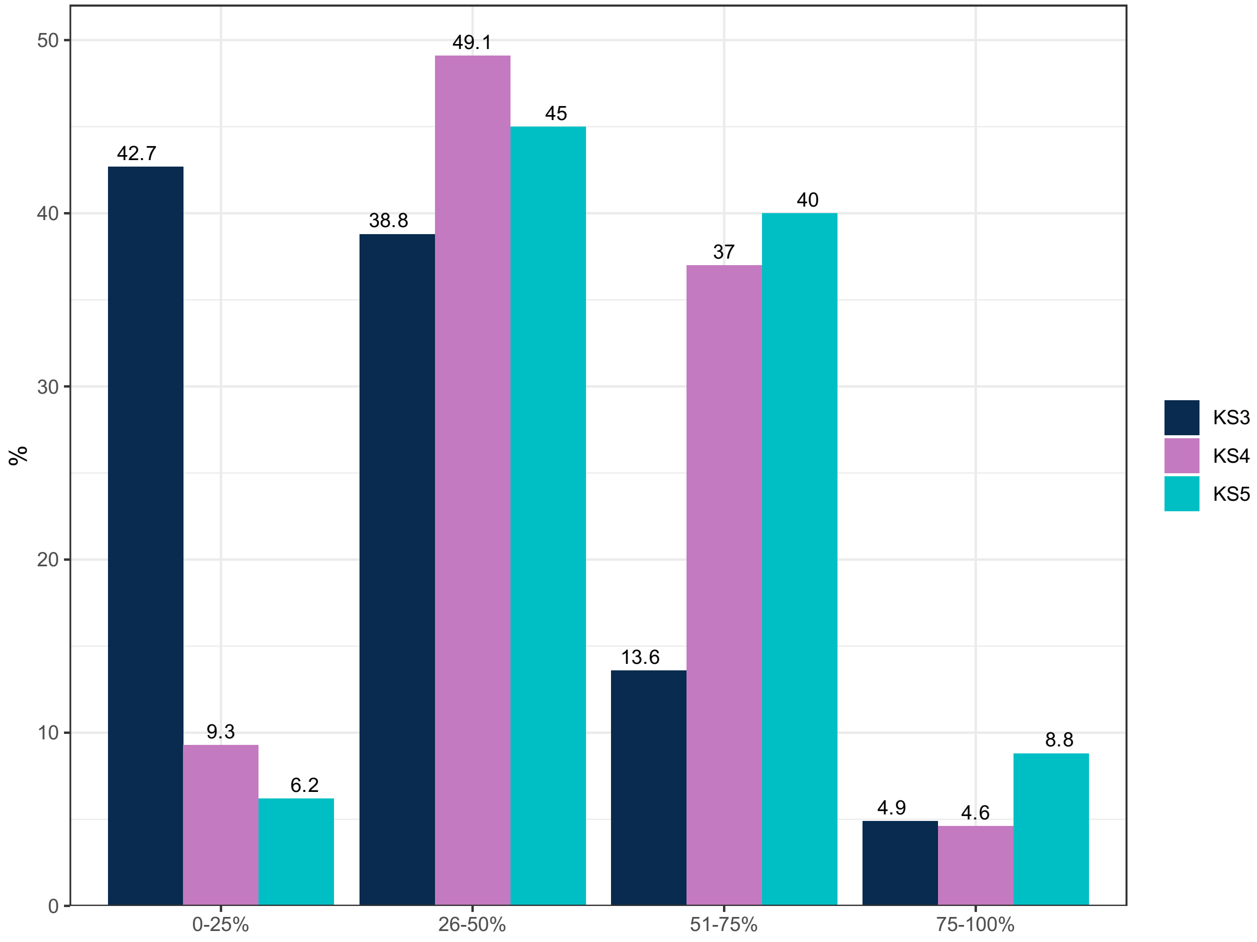


Figure 1: Percentage of programming taught per key stage

When looking at the amount of lesson time spent programming broken down by school type in Figure 2, we can see that state-maintained schools are more likely to spend 0-25% of lesson time at KS3 on programming, whereas independent schools are more likely to spend 26-50%. At KS4, although there are differences in the percentages, 26-50% is the most popular amount of lesson time dedicated to programming in both state maintained and independent schools. At KS5, there is roughly an equal split among independent schools between spending 26-50% and 51-

75%. For state-maintained schools and colleges slightly more spend 26-50% than those that spend 51-75% of lesson time.

When looking at the gender of the teacher delivering programming lessons, Figure 3 shows that there is very little difference in delivery pattern in KS3. However, there are differences in KS4, where 45.4% male teachers had programming in 51-100% of their lessons, compared to 39.5% of female teachers. This might be a reflection of the lessons that different teachers deliver, with potentially more programming lessons being taught by male teachers.

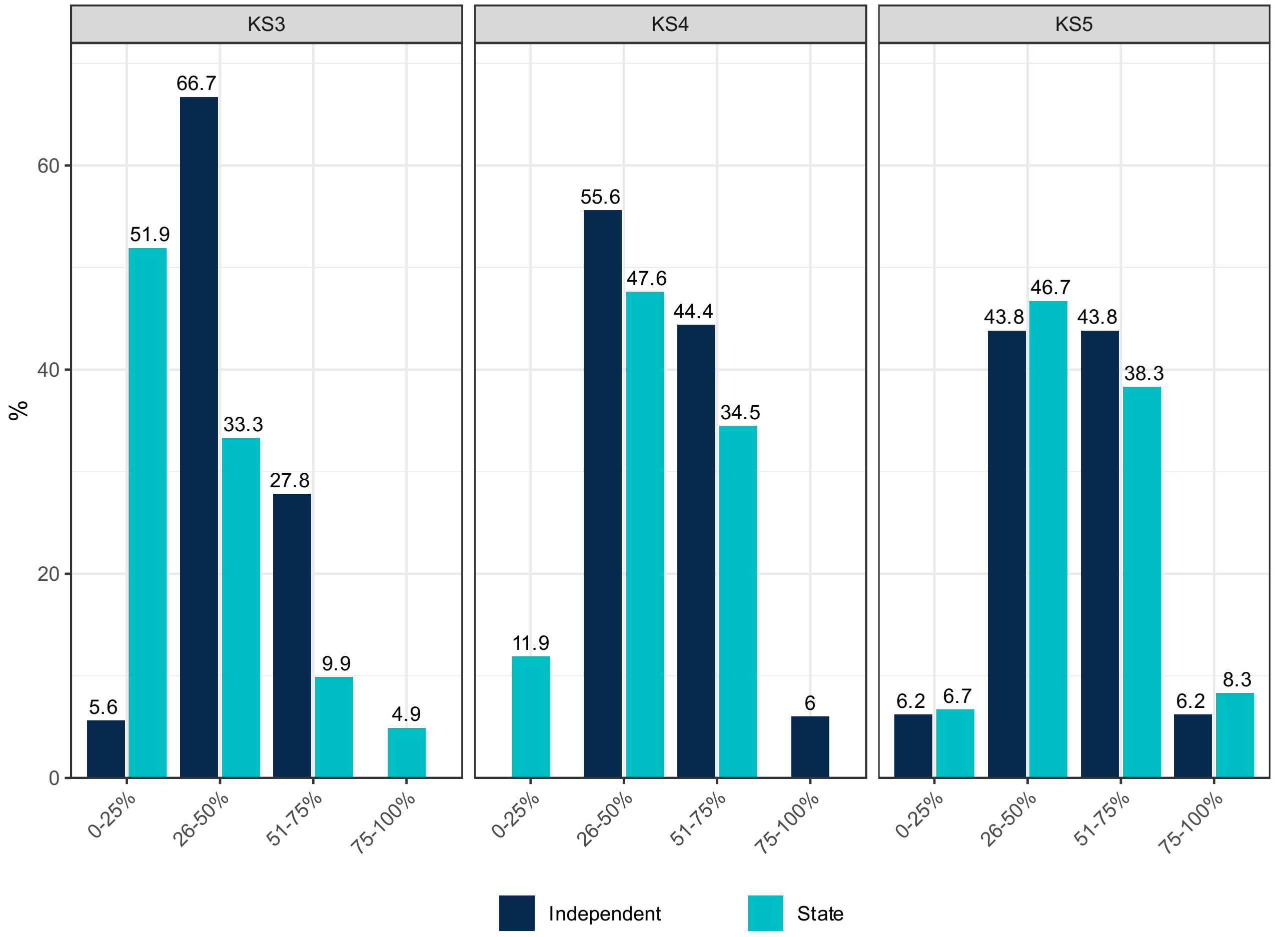


Figure 2: Percentage of programming taught per key stage and school type

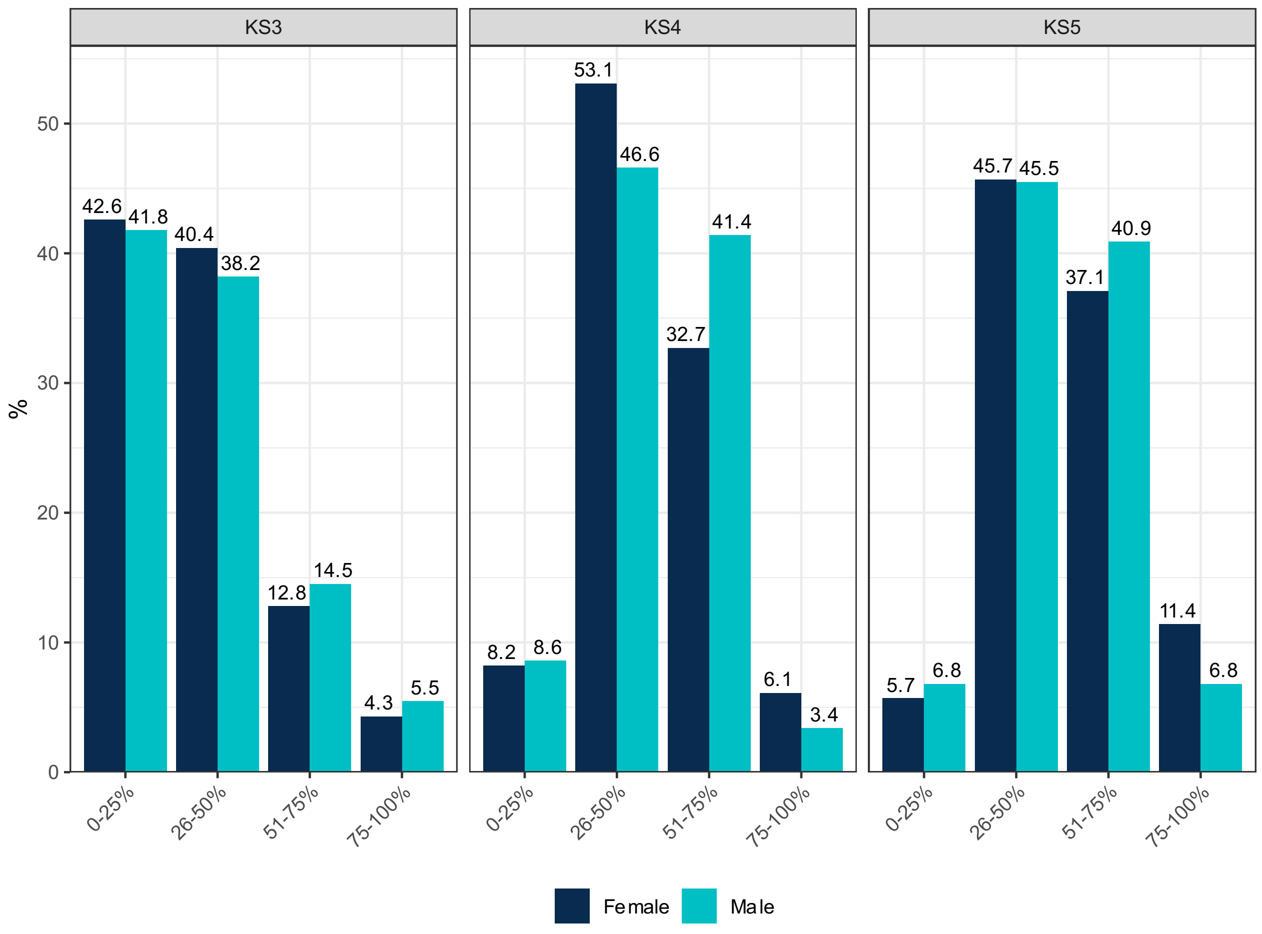


Figure 3: Percentage of programming taught per key stage and teacher gender

In 2017 Ofqual, the exam regulator for England, announced that they would be withdrawing the non-examination assessment from GCSE Computer Science (Ofqual, 2017). Exams for the largest two exam boards, OCR and AQA, are now sat entirely on paper and anecdotal evidence suggests that the amount of on-screen programming to support the exam has decreased [5]. However, this may change in the coming years as there is a drive to offer digital versions of the GCSE Computer Science examinations (OCR, 2023).

We asked participants if having an onscreen programming exam would change the amount of lesson time they dedicated to programming. Figure 4 shows that around 60% of teachers said they would not change the amount of time they spend and almost 40% said they would increase the time dedicated to it.

[5] Anecdotal evidence from school visits. Pearson/Edexcel now offer an onscreen programming exam.

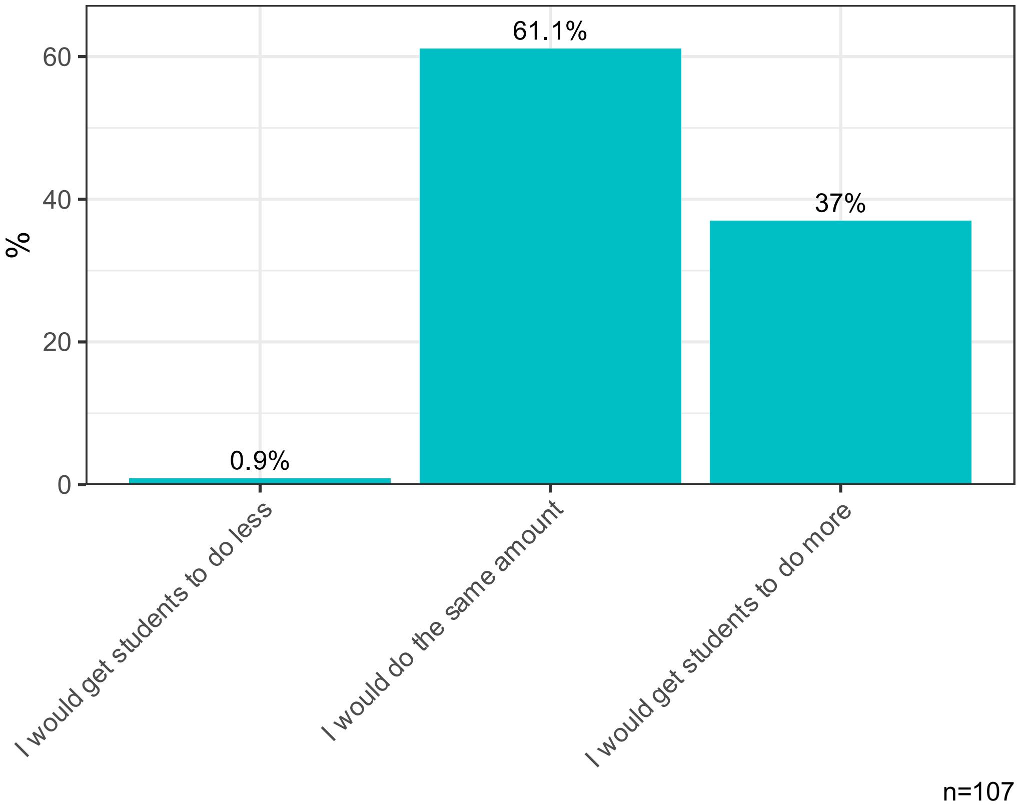


Figure 4: Impact of exam on programming time allocation

Pedagogical approaches

The survey participants were asked which pedagogical approaches they employ when teaching programming. Figure 5 shows that the approach that is used most often is PRIMM (Predict, Run, Investigate, Modify Make) (Sentance, Waite, and Kallia 2019) with 41.7% of teachers stating they use it often and 44.4% saying they use it sometimes. Over half of teachers also reported that they use a number of other approaches sometimes, such as: Pair Programming, Project-Based Learning, Program Tracing and Unplugged Computing.

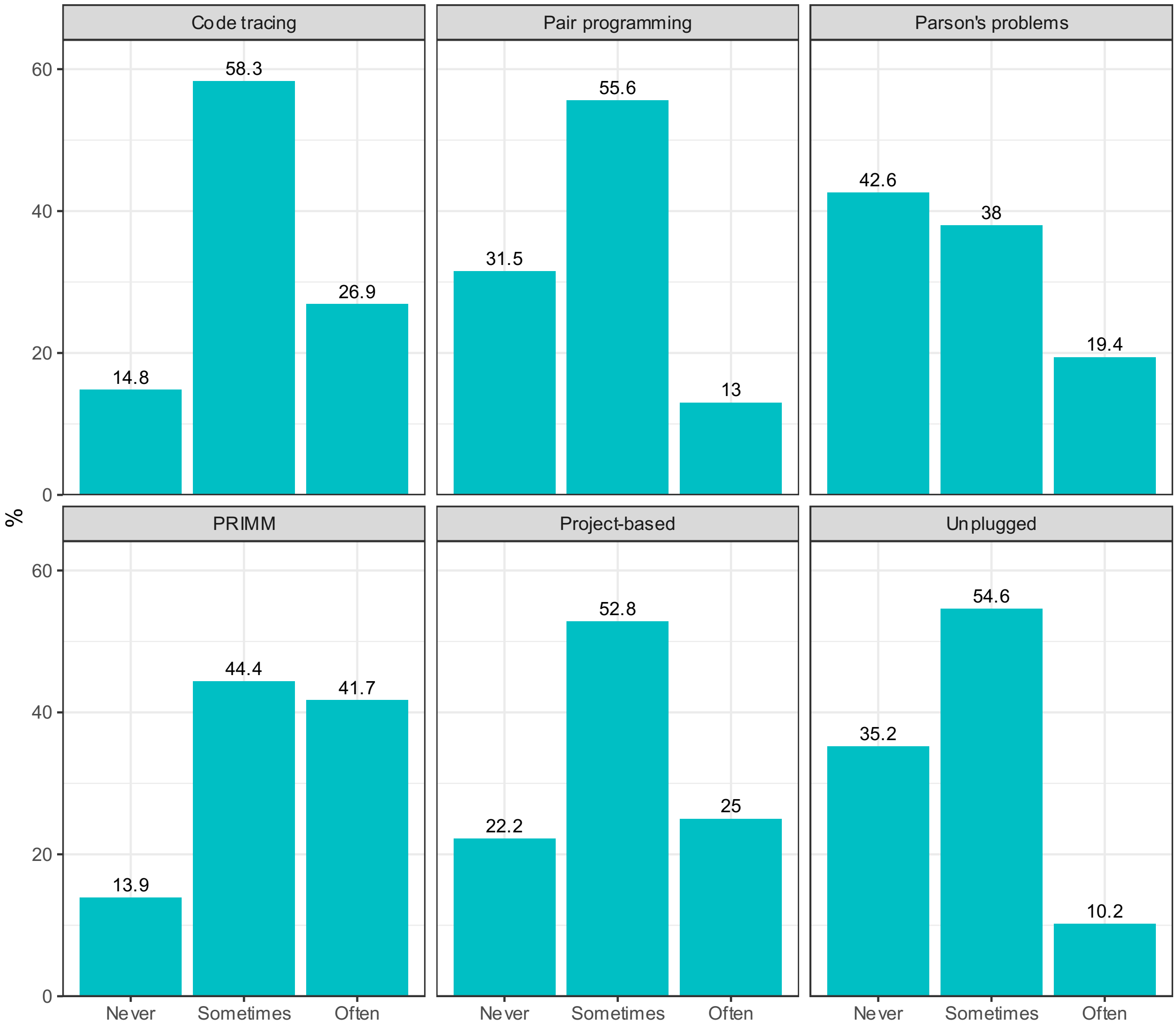


Figure 5: Programming pedagogies employed

When looking at the responses broken down by school type as shown in Figure 6, we can see that teachers in independent schools are more likely to often use PRIMM and Parson's problems whereas state schools are more likely to often use code tracing project-based learning.

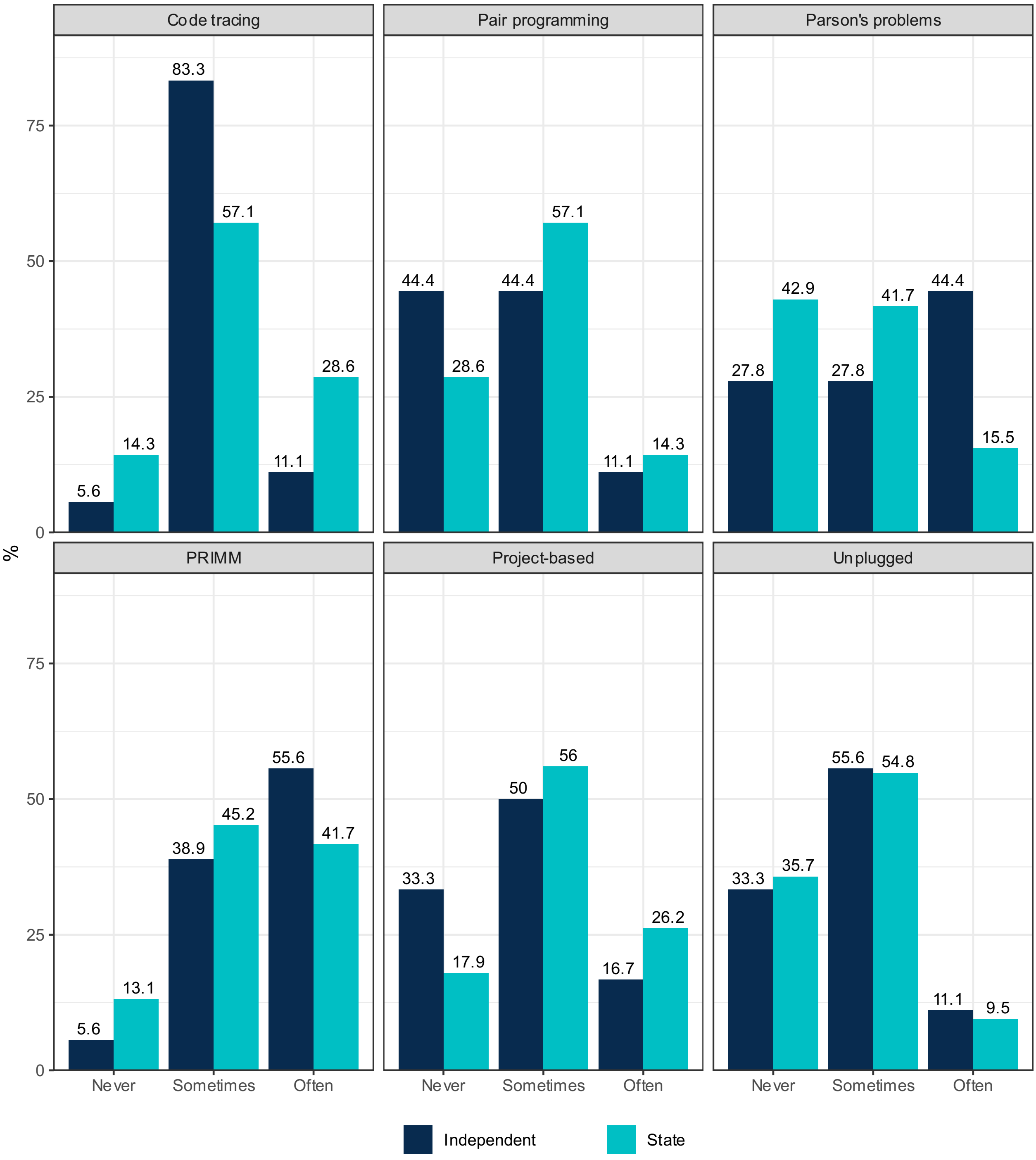


Figure 6: Programming pedagogies employed by school type

When looking at the responses broken down by gender, Figure 7 shows that 51% of female teachers were *often* employing PRIMM in their teaching, compared to just 34.5% of male teachers. Parson's problems were used *often* by 24.1% of male teachers compared to 14.3% of female teachers.

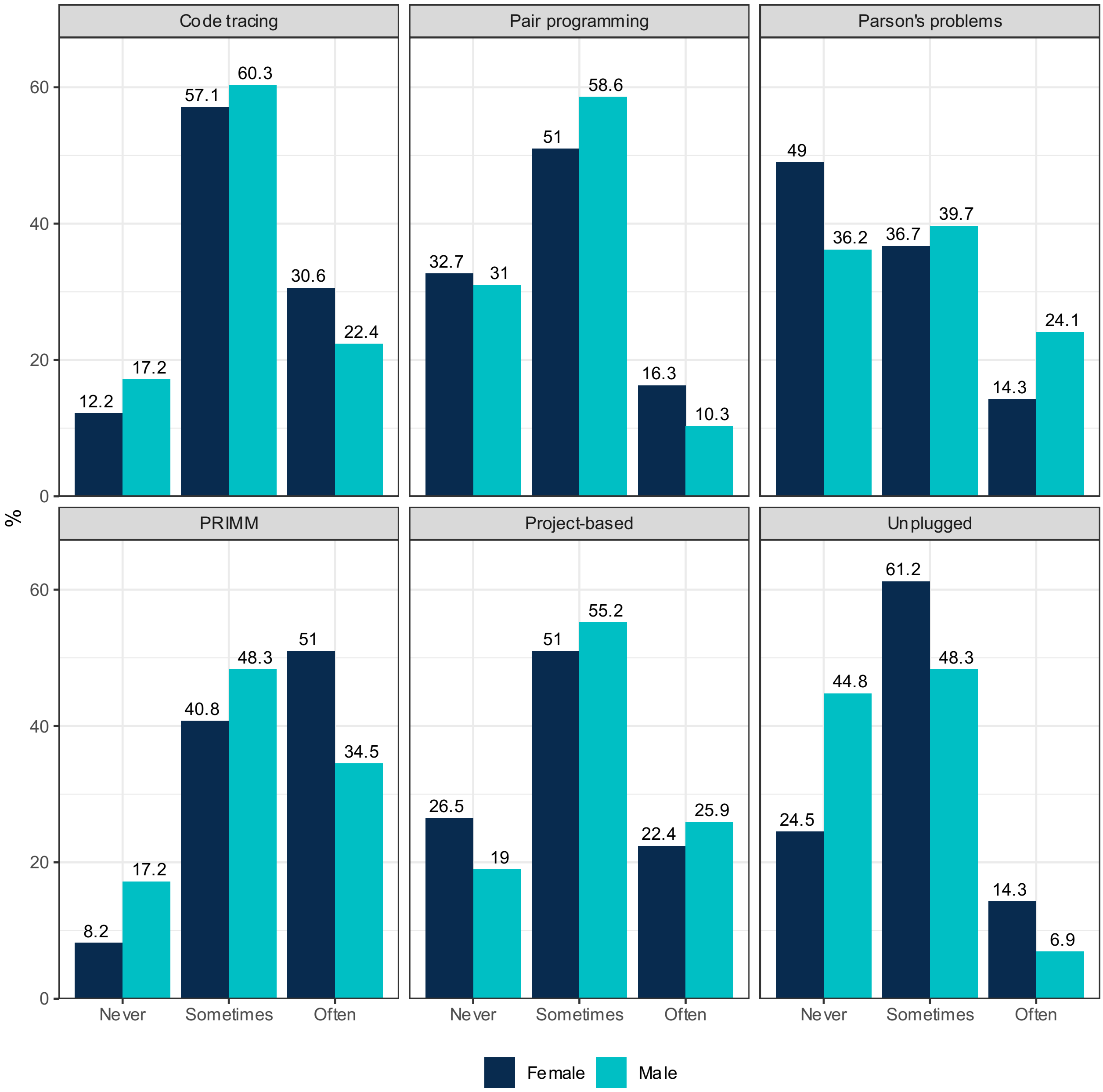


Figure 7: Programming pedagogies employed by teacher gender

Highly able programmers

When we asked teachers if they had ever taught a student who was a better programmer than them, the vast majority, 81%, said that they had. Figure 8 shows that the percentage differed between male and female teachers, with 85.7% of female teachers saying they had met such students, compared to 75.9% of male teachers.

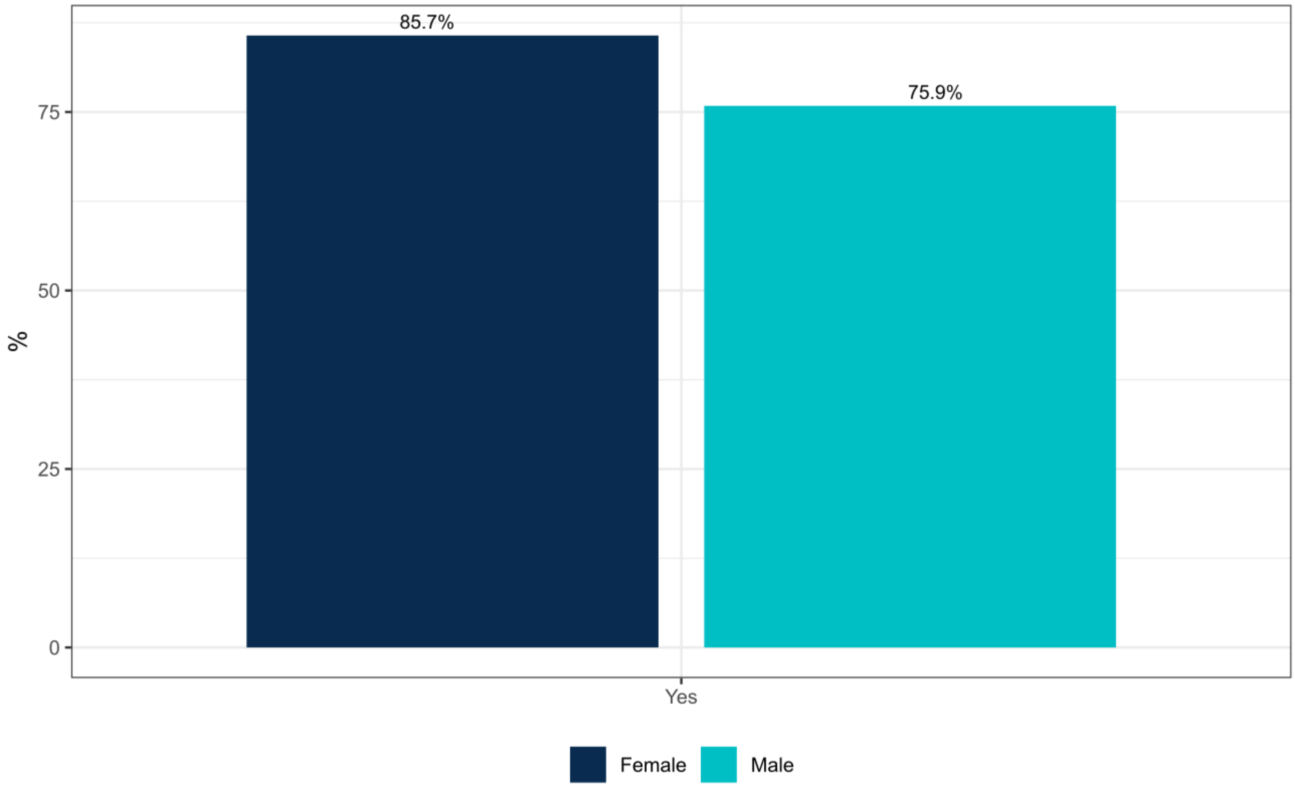


Figure 8: Would you say that you teach/have taught students that are better programmers than you, by teacher gender

We also asked teachers to answer some questions regarding the best programmer they currently teach. Figure 9 shows that 76.2% of learners described as best programmers are male and 22.6% female. This distribution is roughly in line with the national uptake of GCSE Computer Science which was 21% female in 2023 (JCQ, 2023). When asked whether their best programmer was on their school's highly able pupils (or equivalent) register, 59% of teachers confirmed that they were, as shown in Figure 10.

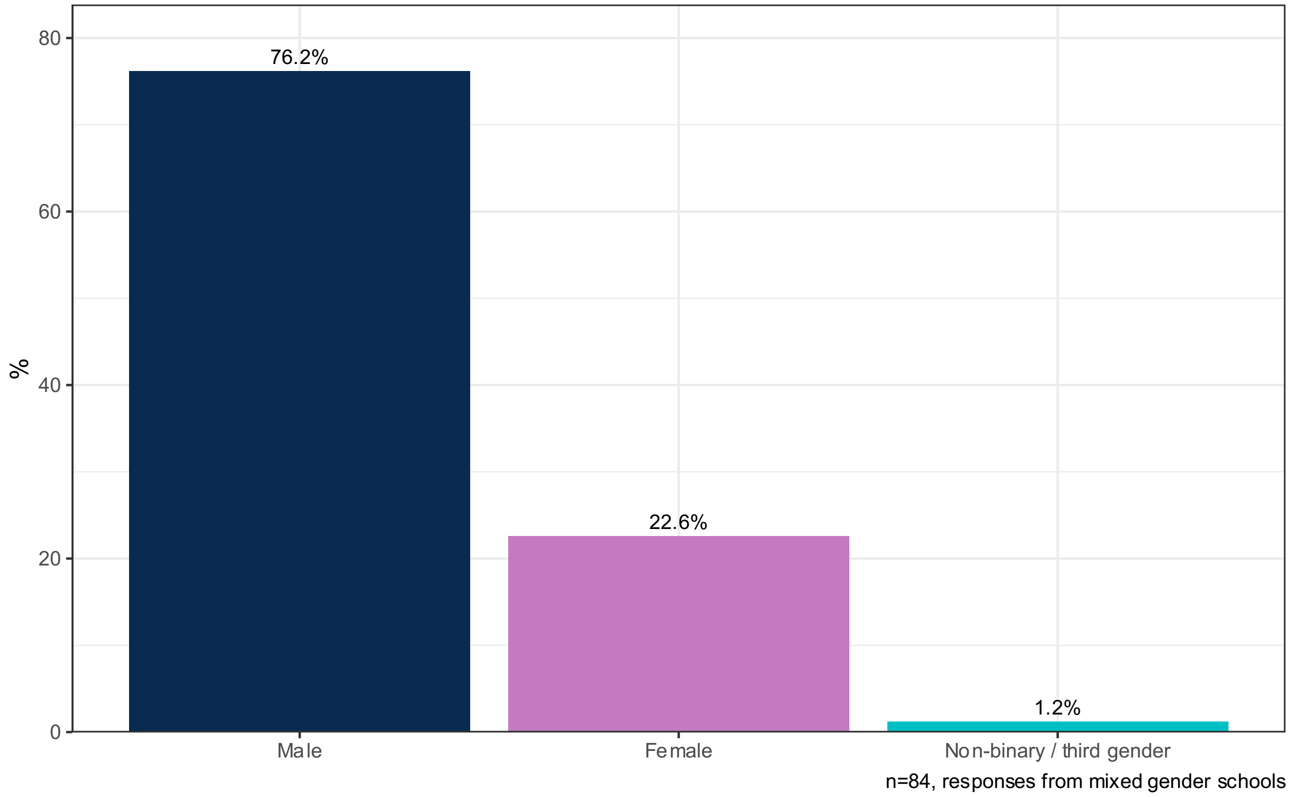


Figure 9: Gender of best programmers

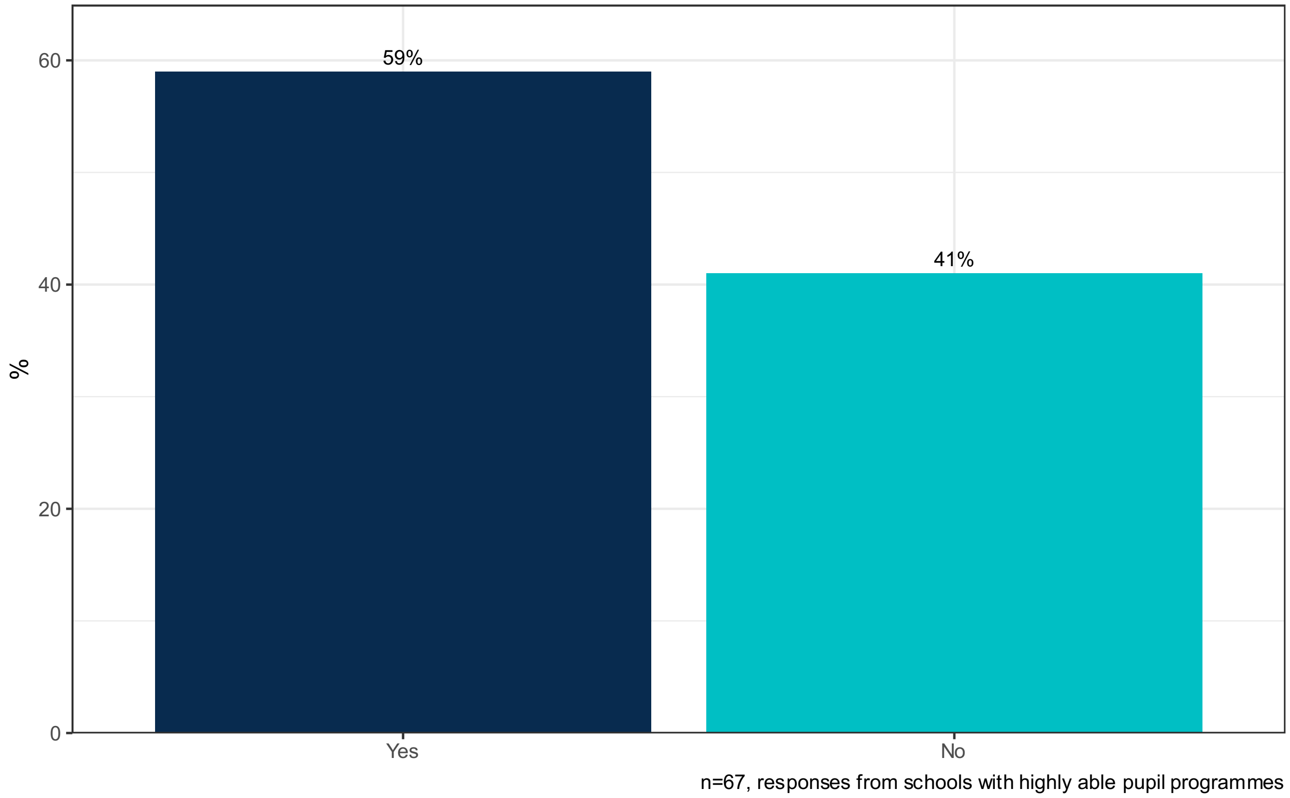


Figure 10: % of best programmers on highly able pupils register (or equivalent)

Extra-curricular programming opportunities

In order to gain a clearer picture of the extra-curricular programming opportunities that students are offered, we asked the participants whether they run any Computing clubs or support their students in entering competitions.

Computing related clubs

As can be seen in Figure 11, a high proportion of all schools run Computing related clubs, however, non-selective state schools have the lowest representation with 72% of providers running clubs.

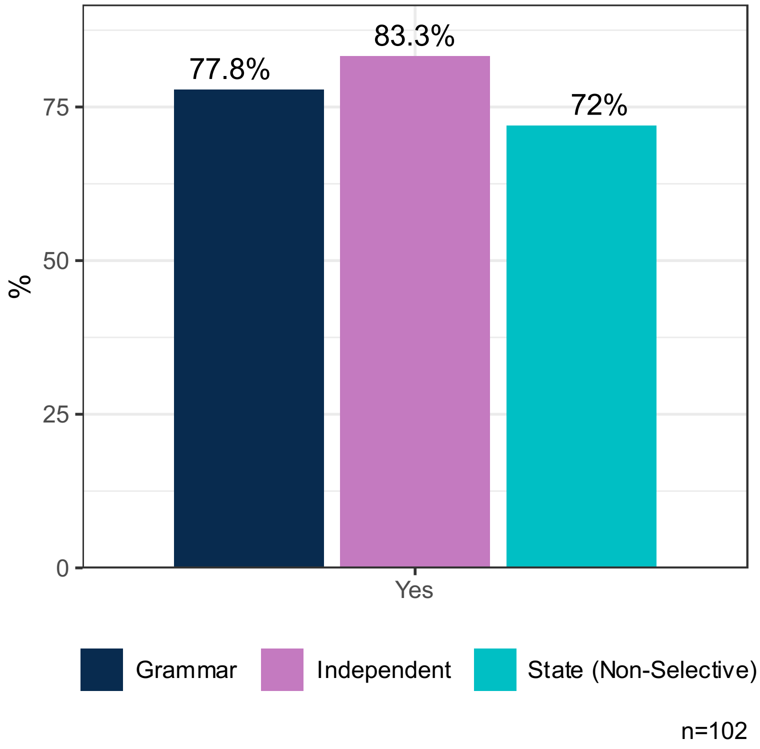


Figure 11: Do you run any computing related clubs?

The participants which indicated that they do run Computing clubs were asked to select the topics they cover. Although not all Computing clubs will necessarily feature programming, Table 6 indicates that the majority do with 69.6% featuring text-based programming and 63.3% featuring block-based.

Table 6: Computing club topics

Club	n	All schools[1]	Schools with clubs[2]
Textual programming	55	50.9%	69.6%
Block-based programming	50	46.3%	63.3%
Homework / revision	43	39.8%	54.4%
Robotics	36	33.3%	45.6%
Hardware	15	13.9%	19.0%
Music creation	3	2.8%	3.8%

[1]All schools n = 108

[2]Schools with clubs n = 79

When looking at the types of clubs offered by school type, we see in Table 7 that grammar schools are more likely to offer textual programming clubs than other types of school and independent schools are more likely to offer robotics clubs.

Table 7: Computing club topics by school type

	State			Private / independent			Grammar		
Club type	**n**	**with clubs**	**all**	**n**	**with clubs**	**all**	**n**	**with clubs**	**all**
Block-based programming	36	67%	48%	9	60%	50%	3	43%	33%
Textual programming	35	65%	47%	10	67%	56%	7	100%	78%
Homework / revision	28	52%	37%	10	67%	56%	5	71%	56%
Robotics	23	43%	31%	10	67%	56%	3	43%	33%
Hardware	11	20%	15%	3	20%	17%	1	14%	11%
Music creation	2	4%	3%	1	7%	6%	-	-	-

Computing competitions

Our data shows that 74% schools and colleges participating in the survey support students in entering external competitions. Figure 12 shows that *all* participating independent schools support students entering computing competitions compared to 70.7% of comprehensives and 77.8% of grammar schools.

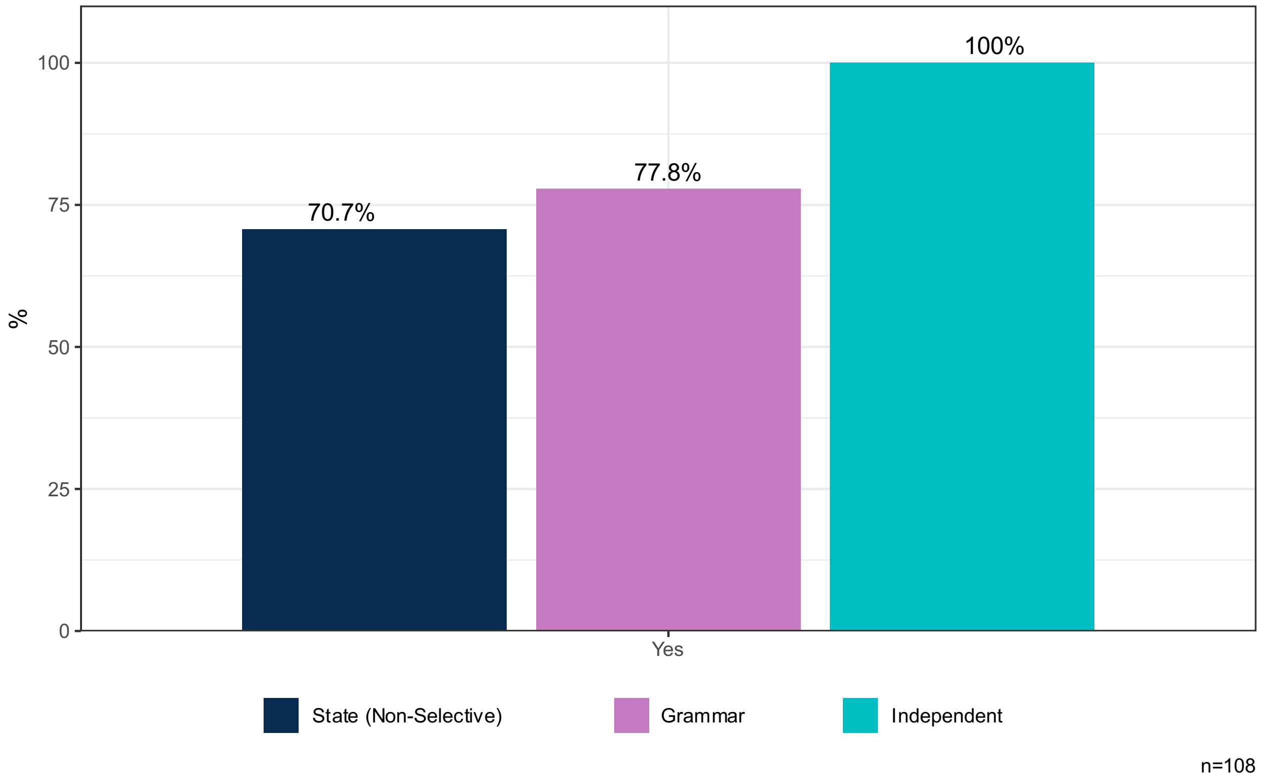


Figure 12: Do you run/support any computing competitions?

Table 8 shows that the most popular competition is Bebras with 79.5% of schools that support a competition running it.

Table 8: Supporting students entering competitions

Competition	n	All schools[1]	Schools with competitions[2]
Bebras	62	57.4%	79.5%
Advent of code	29	26.9%	37.2%
Informatics Olympiad	17	15.7%	21.8%
Perse	14	13.0%	17.9%
Young Games Developer of the Year	9	8.3%	11.5%
Cyber First Girls	8	7.4%	10.3%
Lego League	7	6.5%	9.0%
Vex Robotics	5	4.6%	6.4%
Young Animator of the Year	5	4.6%	6.4%
Student Robotics (Southampton)	4	3.7%	5.1%
Uni of Oxford Computing Challenge[3]	3	2.8%	3.8%
AWS	2	1.9%	2.6%
BAO	2	1.9%	2.6%
National Cipher Challenge	2	1.9%	2.6%

Other competitions with only one school reporting them were: Cipher Challenge; Code Wars; Cyber Explorers; Cybercenturion; DressCode; First Robotics; First Tech Challenge; Hackathon; Immersive Labs; PA Consulting Raspberry Pi; Samsung Solve for Tomorrow; School/Trust Specific; UKLO; Your Voice is Power

[1] All schools n = 108

[2] Schools with competitions n = 78

[3] Only students who achieved a top 10% score in the Bebras Challenge are invited to take part.

If we compare the responses between state-maintained and independent schools in Table 9, it can be seen that entries to the Bebras Challenge and Advent of Code remain high for both settings. However, the Informatics Olympiad and Perse School Team Coding Challenge are twice as popular for independent schools when compared with state maintained. On the other hand, other competitions, such as the Young Game Designer and Young Animator of the Year competitions, are more popular with state-maintained schools.

Table 9: Computing competitions by school type

Competition	State			Private / independent		
	n	% who enter comps (n=60)	% of all (n=84)	n	% who enter comps (n=18)	% of all (n=18)
Bebras	45	75.0	53.6	17	94.4	94.4
Advent of code	21	35.0	25.0	8	44.4	44.4
Informatics Olympiad	10	16.7	11.9	7	38.9	38.9
Perse	8	13.3	9.5	6	33.3	33.3
Young Games Developer of the Year	8	13.3	9.5	1	5.6	5.6
Cyber First Girls	6	10.0	7.1	2	11.1	11.1
Lego League	5	8.3	6.0	2	11.1	11.1
Young Animator of the Year	4	6.7	4.8	1	5.6	5.6
Vex Robotics	3	5.0	3.6	2	11.1	11.1
AWS DeepRacer	2	3.3	2.4	-	-	-
Uni of Oxford Computing Challenge[1]	1	1.7	1.2	2	11.1	11.1
Student Robotics (Southampton)	-	-	-	4	22.2	22.2
BAO	-	-	-	2	11.1	11.1

Other competitions with only one state school reporting them were: Code Wars; Cyber Explorers; Cybercenturion; DressCode; First Robotics; First Tech Challenge; Hackathon; Immersive Labs; National Cipher Challenge; PA Consulting Raspberry Pi; Samsung Solve for Tomorrow; School/Trust Specific; Your Voice is Power

Other competitions with only one private school reporting them were: National Cipher Challenge; Cipher Challenge; UKLO

[1] Only students who achieved a top 10% score in the Bebras Challenge are invited to take part.

If we compare the types of competitions entered between state maintained and independent schools shown in Table 10, it can be seen that game design and animation competitions are more popular with state-maintained schools, whereas to robotics and cyber security competitions are much more popular with independent schools. Competitions focusing on programming and algorithms are popular choice for independent schools with 66.7% supporting their students entering them, however less than half of state-maintained schools participate in these competitions.

Table 10: Type of competition entered by school type

	State			Private / independent		
Competition type	n	% who enter comps (n=60)	% of all (n=84)	n	% who enter comps (n=18)	% of all (n=18)
Computational thinking	45	75.0	53.6	17	94.4	94.4
Programming & algorithms	29	48.3	34.5	12	66.7	66.7
Game design and animation	11	18.3	13.1	2	11.1	11.1
Robotics	8	13.3	9.5	8	44.4	44.4
Cyber security	6	10.0	7.1	4	22.2	22.2
AI and ML	2	3.3	2.4	-	-	-

The computational thinking category only includes the Bebras competition.

Competitions focusing on programming and algorithms included: Advent of Code, Informatics Olympiad, Perse School, Oxford University Computing Challenge and British Algorithmic Olympiad.

The game design and animation category included: Young Games Developer of the Year and Young Animator of the Year.

The robotics category included: Lego League, VEX Robotics and Student Robotics (Southampton University).

The cyber security category included: CyberFirst and the National Cipher Challenge.

The AI and machine learning category only includes AWS DeepRacer.

Discussion

Programming within the secondary curriculum

Python continues to be the most popular programming language employed by teachers in secondary schools in England. Additionally, block-based languages such as Scratch are used by over half of schools at KS3. Their use drops to less than 3% at KS4 and 5, however this is most likely due to the main examination boards for GCSE and A-Level Computer Science not currently accepting responses in block-based languages in their assessments. Therefore, this does not mean that all teachers believe that these languages are not suitable for teaching programming at these levels. The use of other languages such as C#, Java and Javascript increases at KS5. This is not surprising, as at this level, learners work on their own original complex programming project and therefore they need to select the most appropriate programming languages and frameworks for their purpose.

It was interesting to explore the amount of lesson time that schools dedicate to teaching programming at different key stages. There was quite a bit of variation, with the majority of schools spending 0-50% of lesson time on programming at KS3. It being less than 50% makes sense given the scope of the Computing National Curriculum (DfE 2014) covers three broad areas: Computer Science, Information Technology and Digital Literacy. At KS4 in most schools, students will be working towards the GCSE which just focuses on Computer Science. Therefore, it makes sense that the amount of time dedicated to programming increases, with the majority of schools falling in the 26-75% range and 49.1% in the 26-50% range. A similar picture is seen at KS5 at which point most students take their Computer Science A-Level. It is interesting to note that independent schools are more likely to dedicate 26-50% of lesson time to programming at KS3 whereas state-maintained schools are more likely to spend 0-25%.

The participants were also asked whether having an on-screen programming exam at GCSE would change the amount of lesson time dedicated to programming. Currently the two most popular examination boards for Computer Science in the UK assess programming at GCSE through a written paper. 61.1% of teachers said that the amount of time dedicated to programming would stay the same and 37% said it would increase. This is an important consideration given the OCR examination board recently announced that their GCSE Computer Science qualification will be 'the UK's first major high-stakes qualification set to be fully assessed on screen' (OCR, 2023).

PRIMM has been shown to be the most popular pedagogical approach to teaching programming in secondary education, with over 40% of teachers saying they use it often. Over half of teachers said they sometimes use other approaches such as pair programming, project-based learning, program tracing and unplugged computing. Interestingly, 42.6% of teachers had never used Parson's problems which makes us reflect on whether this is the case or if they do not know them by this name.

The majority of teachers reported that they have taught a student that is a better programmer than them. Such learners may fit the category of highly able pupils (previously known as gifted and talented) as it has been suggested that these learners 'often know more than the teacher' (NCCA, 2007, p. 20). However, only 59% of teachers reported that the best programmer they currently teach is on their school's highly able pupils register (or equivalent).

Extra-Curricular programming opportunities

While it is positive to see that the majority of schools are offering and supporting extra-curricular opportunities, 28% of state-maintained non-selective schools currently do not. This is particularly relevant as Kemp et al. (2024c) found that girls attending state-maintained schools were more likely to study GCSE Computer Science if they were taking part in digital making activities outside curriculum time. However, there were large gendered differences in the frequency and types of digital making undertaken. This reinforces the importance of providing support and opportunities for all students to engage with digital making activities outside curriculum time.

Although Computing clubs do not necessarily involve programming, it seems that the many do with 69.6% of schools running at least one club that involved text-based programming and 63.3% included block-based provision. The difference between state-maintained and independent schools in terms of offering any type of Computing related club is relatively small, however, independent schools are more likely to offer robotics clubs, a type of club that potentially comes with significant hardware costs, raising questions about how socio-economic factors might be influencing access to this type of activity.

In terms of competitions, *all* independent schools reported that they support their learners in entering them and most state-maintained schools do the same. The Bebras challenge, which focuses on computational thinking, is by far the most popular computing competition, with 79.5% of schools taking part. Independent schools are much more likely to enter their students into programming focused and robotics competitions. Whereas state-maintained schools are more likely to enter their students into game design and animation competitions. Why this is the case warrants further investigation as it could have implications for the equity of computing competitions for students from different backgrounds. Another area for potential investigation is the exploration of the destinations of students who take part in programming focused competitions compared with those that participate in applied computing competitions which feature aspects of programming.

Conclusion

The results of this survey highlight the diverse approaches to teaching programming in secondary education across England. Python remains the dominant language, especially at KS4 and KS5, while block-based languages like Scratch are popular at KS3 but decline in later stages likely due to the way that programming is currently examined. Programming instruction time varies significantly across key stages, with independent schools generally dedicating more time than state-maintained schools. The potential shift toward on-screen programming exams could further influence this distribution. Pedagogically, PRIMM stands out as a favoured method, though other strategies like code tracing and project-based learning are also frequently employed.

Extra-curricular opportunities, such as clubs and competitions, are prevalent across schools, providing important platforms for students to engage with programming and other forms of digital making outside the classroom. However, differences in opportunities between state-maintained and independent schools, particularly in robotics competitions, highlight disparities that may affect students' experiences and opportunities. Further investigation into the reasons behind these discrepancies, as well as the impact of participation in different types of competitions, could offer insights into promoting greater equity in computing education.

Acknowledgements

This research was generously funded by the King's College London Centre for Research in Education based within the School of Education, Communication & Society.

Bibliography

Blackwell, Alan F. 2002. "What Is Programming?" In *PPIG*, 14:204–18. Citeseer.

DfE. 2014. "The National Curriculum in England: Framework Document." Department for Education, London.

IFF Research. 2023. "School and College Panel - June 2023." https://assets.publishing.service.gov.uk/media/6537e3e65e47a5000d989912/School_and_College_Panel_June_2023.pdf.

JCQ. 2023. "GCSE (Full Course) Results - August 2023." https://www.jcq.org.uk/wp-content/uploads/2023/08/GCSE-Full-Course-Results-Summer-2023.pdf.

Kemp, Peter E.J., Wong, Billy, Hamer, Jessica, Copsey-Blake, Meghan. 2024a. "Computing Provision Report: English School Data 2011-2023." https://scari.sites.er.kcl.ac.uk/cpre/.

Kemp, Peter E.J., Wong, Billy, Hamer, Jessica, Copsey-Blake, Meghan. 2024b. *SCARI Computing Report - teacher workforce*. King's College London. Available at: http://kcl.ac.uk/scaricomputing

Kemp, Peter E.J., Wong, Billy, Hamer, Jessica, Copsey-Blake, Meghan. 2024c. "The Future of Computing Education - Considerations for Policy, Curriculum and Practice." https://www.kcl.ac.uk/ecs/assets/kcl-scari-computing.pdf.

NCCA. 2007. *Exceptionally Able Students: Draft Guidelines for Teachers.* www.ncca.ie

OCR 2023. "OCR to offer first GCSE with fully digital exams" https://www.ocr.org.uk/news/ocr-to-offer-first-gcse-with-fully-digital-exams/

Ofqual (2017). Consultation on revised assessment arrangements for GCSE computer science. Ofqual.

Royal Society. 2017. "After the Reboot: Computing Education in UK Schools." *Policy Report*.

Sentance, Sue, Jane Waite, and Maria Kallia. 2019. "Teaching Computer Programming with PRIMM: A Sociocultural Perspective." *Computer Science Education* 29 (2-3): 136–76.