How is programming taught in code clubs? Exploring the experiences and gender perceptions of code club teachers

Efthimia Aivaloglou
Leiden Institute of Advanced Computer Science
Open University of the Netherlands
e.aivaloglou@liacs.leidenuniv.nl

Felienne Hermans
Leiden Institute of Advanced Computer Science
f.f.j.hermans@liacs.leidenuniv.nl

ABSTRACT

An increasing number of school-age children currently learns programming at after-school programming clubs, also referred to as code clubs. Some clubs are part of international programs like CoderDojo or Code Club, while others are independent. This work investigates how code clubs are organized in terms of participants, lesson material, style and assessments by means of a survey. It also explores the teachers' experiences and perceptions related to gender differences among their code club students and the learning difficulties that they identify in their classes. The survey results shed light on the differences between code clubs and school lessons. Code club teachers often have a computer science background and no education experience. Moreover, motivation and commitment are rarely identified as learning barriers for code club students, whereas debugging, error messages and abstract thinking are the most commonly reported difficulties. With respect to gender differences, boys are commonly perceived as being more confident and girls as being more persistent and able to concentrate better. Gender differences are also found in the preferred type of projects, the responsiveness to instruction and collaboration skills.

CCS CONCEPTS

• **Social and professional topics** → *Computing education*; *Gender*.

KEYWORDS

programming education, K-12, after-school programming clubs, code clubs, learning barriers, gender differences

ACM Reference Format:
Efthimia Aivaloglou and Felienne Hermans. 2019. How is programming taught in code clubs? Exploring the experiences and gender perceptions of code club teachers. In *19th Koli Calling International Conference on Computing Education Research (Koli Calling '19), November 21–24, 2019, Koli, Finland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3364510.3364514

1 INTRODUCTION

Programming is a popular extracurricular activity for school-age children. International programs like CoderDojo or Code Club are supporting an increasing number of after-school programming clubs; Code Club alone currently reports having 13 thousand clubs with 180 thousand students in 160 countries[1]. There also exist code clubs targeted to female learners, for example the ones supported by the Girls Who Code program. Programming classes are also given in independent code clubs, coding summer camps, workshops in libraries and museums, as well as various other venues.

Code club lessons are different from in-school programming classes in terms of setting and lesson material, since these are not being dictated by national curricula and can be more informal. The learning environment in code clubs has already been found to affect student emotions [18] and motivation [4], where freedom of choice has been identified as a motivational factor for code club students. This is different than the case of school students, where motivation had been identified as a learning barrier by school teachers [9].

The goal of this paper is to explore how programming is taught at code clubs through the experiences and perceptions of code club teachers. We investigate if the learning barriers that have been reported by school teachers in programming, including motivation, committent and abstraction capacity [9], also apply to students of code clubs. Perceptions of gender differences have also been investigated among school teachers and structuredness, self-confidence and scientific curiosity have emerged as the most commonly perceived differences between female and male students [11]. In this study we aim to determine if the gender perceptions of school teachers also apply to the case of code clubs. Finally, we are interested in exploring whether the age, gender and education background of the teachers affect these perceptions.

Specifically, we are examining the following research questions:

- **RQ1** How is programming taught at code clubs in terms of participants, lesson material and style, and assessments?
- **RQ2** What learning barriers in programming do the teachers of code clubs identify?
- **RQ3** What are the teachers' perceptions of gender differences among their code club students? Are these affected by the teachers' age, gender and education background?

To answer our research questions, we conducted an exploratory survey where we invite code club teachers to answer to a combination of closed and open-text questions.

The results from the survey and the qualitative analysis of the open-text responses suggest that (1) the teaching material and practices vary between different code clubs and different programs, but (2) plenary sessions and summative assessments are rare. Also, (3) the most commonly identified learning difficulties relate to debugging, error messages and abstract thinking, while commitment and

Koli Calling '19, November 21–24, 2019, Koli, Finland

ACM ISBN 978-1-4503-7715-7/19/11…$15.00
https://doi.org/10.1145/3364510.3364514

[1] https://codeclub.org/, accessed: August 1, 2019

motivation are rarely reported as learning barriers for code club students. Gender differences are perceived by code club teachers, with (4) boys in code clubs being perceived as being more confident and having more familiarity and prior knowledge of programming, whereas (5) persistence, concentration, responsiveness to instruction, collaboration skills, grit and structuredness are considered increased for girls.

2 BACKGROUND AND RELATED WORK

Computer science education research is mainly focused on the context of school classrooms [13] and numerous studies have been carried out on teaching programming concepts in classroom settings [16]. Alongside these there are works reporting on teaching programming outside the classroom, as an extracurricular activity, for example at summer coding camps [6, 10], online courses that students can follow at home [14], after-school clubhouses [17], Code Clubs [23], or CoderDojos [22].

Code Club was founded in 2012 as network of after-school clubs in primary schools, and survey results after its first year of operation have been reported by Smith *et al.* [23]. CoderDojo was founded soon after, in 2012, as a movement of community-based programming clubs for students from seven to seventeen years old. The skills gained by CoderDojo participants in terms of knowledge and understanding, communication and learning skills, autonomy and responsibility have been explored in [22]. Code Club and CoderDojo conduct annual surveys for the people who teach at their code clubs, that offer rich information on their operation and teacher experiences [7, 8]. This information includes teachers' perception on the acquisition of programming skills by the students, the number of students, the rate of female students, the teaching material, and regional differences between code clubs.

The effect of the learning environment in code clubs, and specifically CoderDojos, on student emotions has been examined in [18]. The interactive and informal nature of CoderDojo sessions were found to make social emotions such as admiration, stress, or excitement particularity important, because they can directly influence other factors like motivation and performance. Butler *et al.* focused on motivation aspects in CoderDojos [4]. They explored students' motivation to attend CoderDojo, as well as their perceptions of the experience, and identified five characteristics of CoderDojo that mediate attendees' motivation: provision of support, scope for creativity, freedom of choice, provision of challenge, and friendships. The effect of mentoring in summer camps was studied in [6], where it was found that the self-efficacy and interest in computer science of middle-school participants was affected by their relatability to their near-peer mentors of high school age.

Learning barriers. The increased motivation of code club students contrasts with the case of school students, where motivation has been identified as a learning barrier for programming classes by Dorn *et al.* [9]. In their study, they explored the perceptions of secondary school teachers and computer science first-year students related to learning barriers for programming. Analysing the teachers' answers to the question *"What do you think are the difficulties for students to learn programming?"*, the most frequently mentioned difficulties related to programming language syntax, commitment, motivation/interest, organizational factors, modelling, algorithms, and abstraction capacity. Error messages and debugging were also found to be reported, but not frequently.

Examining the learning barriers related to programming languages, there are specific programming concepts have been found to be challenging for young learners. Middle school students were found to have problems with concepts related to initialization, variables and concurrency in Scratch [19]. Analysing Scratch projects, Seiter and Foreman found that patterns related to parallelization, conditionals and, especially, variables were under-represented until a certain age [21]. Boolean operators and variables were found to be underutilized in Scratch students' projects in [17]. Programming abstractions related to variables and procedures were also found to be underutilized in a large dataset of Scratch programs in [1].

Gender differences. Several gender differences have been identified among the students of programming classes. Most related to our work is the study of Funke *et al.*, who investigated school teachers' perceptions of gender differences [11]. Analysing the answers of 57 participants to the question *"What are the differences between girls and boys concerning programming?"*, they found that the main perceived differences pertained structuredness, self-confidence, and scientific curiosity. Less commonly perceived differences included interests, learning receptivity, accuracy, previous knowledge, perseverance, creativity, frustration, teamwork and evolution. In a later study, Funke and Geldreich verified the existence of gender differences in primary school children learning Scratch by analysing their programs and finding that the students had created different types of programs using different types of Scratch blocks, with boys favouring the motion-category blocks and girls favouring the looks-category blocks [12].

A different approach for investigating gender differences was adopted in [5], where the analysis of gender differences was based on school students' answers. Answering follow-up questions after programming classes, boys appeared to be more confident, while girls appeared to be less fascinated, adopting a pragmatic approach to programming, to be more influenced by positive and negative feelings in their learning process and motivation, to consider communication very important, and to benefit from collaboration. Gender differences in collaborative programming settings were also studied by Tsan *et al.*, who identified differences related to the quality of the programs produced by groups of 5th grade students of different gender compositions, with all-female groups achieving significantly lower program quality than groups with at least one male [24]. They also identified factors that played an important role in the outcomes, including the students' willingness to engage in design and their collaborative practices. A recent large-scale study on students' mathematics and science test scores has also revealed gender differences, with boys benefiting significantly more from inquiry- and problem-based learning than girls [3].

3 METHODOLOGY

The goal of this study is to understand how programming is taught at code clubs. To answer our research questions, we conducted an empirical evaluation of data that we collected by means of a survey designed for people who teach at code clubs. We used a survey for several reasons: (1) most of the data required for answering our research questions is structured (concerns facts and behaviours), along with opinions and beliefs that were modelled in the survey as open-text questions, (2) a large sample was required to represent the various settings in which code clubs operate globally, (3) we are interested in data frequencies or patterns and (4) self-administered surveys are known to apply less social pressure than interviews, reducing social desirability bias [15], which in our case could apply to the questions about the gender differences, associated with RQ3.

In the following paragraphs we describe the survey design and the process that we used for conducting it and for analysing the collected data.

3.1 Survey design

A total of twelve questions were used for collecting data, while five additional questions concern demographic and profile characteristics of the respondents (gender, age, education background). The survey was designed by the authors. The candidate participants are people who are teaching programming at out-of-school programs or code clubs. Table 1 lists the questions, grouped according to the corresponding research question.[2] All questions that allow for single or multiple choice answers also allow for an open-text response (by selecting the *"other"* option). Q3.1 includes more specific sub-questions.

For the design of the survey, we followed guidelines about the survey introduction, question phrasing and design, and question ordering [15, 20]. In the introduction part of the survey, we introduce the research team and we explain the topic and general goals of the survey, along with the estimated time to complete. In order to make the purpose of our research immediately visible after the survey introduction, the survey starts with the questions that were most directly related to its topic, namely Q1.1. to Q1.9. The survey finishes with Q0.1 to Q0.5, following guidelines about leaving the most personal and the most time-consuming questions for last. None of the questions were compulsory, and there were no pre-selected or pre-filled in answers. The survey was reviewed by eight colleagues who filled it in and reported their observations. Those answers were subsequently removed and the survey was revised to address their comments before being published.

3.2 Data collection

The survey was implemented in Google Forms and was collecting responses from November 2018 until April 2019. Candidate participants were invited through several communication channels, including social networks (messages on Twitter and posts on Facebook groups of code clubs of several countries), messaging channels of code clubs (for example, nation-wide Slack channels of CoderDojo teachers), newsletters of code clubs, as well as direct emails to teachers of code clubs. The invitations clarified that the survey is aimed for people teaching at after school programming clubs, mentioned the time to complete and, for all invitations apart from the ones on Tweeter, described the topic of the research. At least two reminders were sent for each invitation.

In total, we obtained 98 complete responses. The number of answers to each question is reported in the last column of Table 1. 9 out of 17 questions were filled in by all participants, while the last question of the survey (Q0.5) was filled in by 93. The lowest response rate was for open-text question Q3.2, where 54 teachers answered on gender differences not already indicated in question Q3.1. We believe that this is due to Q3.1 already including sub-questions for six commonly reported differences [5, 11, 12, 24].

[2]The survey is available at: https://tinyurl.com/y27xbk4p.

3.3 Data analysis

To answer our research questions, we performed both quantitative analysis of the data collected through the survey, as well as qualitative analysis of the open-text responses.

For RQ1, we statistically analysed the answers in the second group of questions in Table 1. This set of questions includes questions related to how code clubs are organized in terms of lesson frequency and participants (number, ages, gender), as well as questions about the teaching material, style and assessments used. For some of the answers to Q1.1 and Q1.2 we had to do some pre-processing, calculating the average for answers in the form *"15-17 children"*, in which case we ended up using 16 as the average. Two answers to Q1.1 had to be discarded because they reported numbers over 1000 and we assumed that they were mistakenly inserted.

RQ2 is addressed by Q2.1 in the survey. This asks teachers to answer to the question *"What do the students struggle with at your code club?"* by selecting the learning difficulties that they have identified out of a set of 10 difficulties commonly reported in the bibliography [1, 9, 17, 19, 21], or by contributing their own open-text answer. The open-text responses were subsequently coded into categories.

To answer RQ3 about perceptions on gender differences we used the data collected from Q3.1, where teachers were asked to reply to six sub-questions of the type *"Who is more confident?"*, by selecting between *"Boys"*, *"Neutral, but maybe boys"*, *"Neutral"*, *"Neutral, but maybe girls"*, or *"Girls"*. The six sub-questions were selected to represent gender differences reported in the bibliography [5, 11, 12, 24]. The results of the ranking sub-questions were quantitatively analysed. We also looked at individual correlations between the teachers' answers to these questions and their gender, age group, main degree and education experience, which we tested for significance.

In order to capture gender differences of types other than the ones covered by this question, we used the follow-up question Q3.2, which was open-text. We identified several new differences through a qualitative analysis of these open-text responses. Initially, two authors worked independently to assign the responses to appropriate categories. To ensure coding reliability, they then revisited their differences together and merged the resulting category system. The final categories were then used by the same researchers, who independently re-coded the responses. In this final coding the intercoder agreement was investigated and no significant differences were found.

4 RESULTS

4.1 Respondents' profile

Table 2 summarizes the demographics and education profile information of the code club teachers that responded to the survey. Most teachers (54%) are between 31 and 46 years old, and 51% of them are male.

Table 1: Survey questions

RQ	Id	Question	Single Choice/ Multiple/ Rank/ Open	No. responses (open-text)
Profile	Q0.1	What is your gender?	S	98 (0)
	Q0.2	How old are you?	S	98 (0)
	Q0.3	What is the topic of your main degree?	S	98 (29)
	Q0.4	Do you have any education experience?	S	98 (15)
	Q0.5	Which programming languages do you know well enough to program in?	M	93 (8)
RQ1	Q1.1	How many children are in your code club?	O	83 (83)
	Q1.2	What is the age range of the students in your code club?	O	97 (14)
	Q1.3	What it the male/female ratio of your students?	R (10 scales, from 100% male to 100% female)	97 (0)
	Q1.4	Which languages or programming environments do you teach?	M	98 (25)
	Q1.5	How frequent are the sessions at your code club?	S	98 (10)
	Q1.6	Is your code club part of a program?	S	97 (13)
	Q1.7	Do you use a predefined set of lessons (sometimes called a lesson plan)?	M	98 (36)
	Q1.8	What is the style of your teaching in your code club?	M	98 (15)
	Q1.9	Do you use any form of assessment?	M	62 (25)
RQ2	Q2.1	What do the students struggle with at your code club?	M	98 (25)
RQ3	Q3.1	In what ways are boys different than girls in your classes? (sub-questions for confidence, motivation, interest, persistence, understanding, concentration, e.g. "Who is more confident?")	R (5 scales, from boys to girls)	96 (0)
	Q3.2	Which other differences have you noticed between how boys and girls learn programming?	O	54 (54)

Asked about the topic of their main degree, 49% of the teachers responded that it is computer science-related. Only 18% of the teachers reported their main degree being related to teaching or education. Analysing the answers of the 32 teachers that responded having an *"Other"* main degree, we find that 10 have a science-related degree (on physics, mathematics, or chemistry), 4 teachers have an engineering degree, 4 have no degree, and the remaining 14 have degrees in other areas (for example, economics, public administration, librarianship, photography).

Examining the education experience of the teachers, 48% responded having none and 37% responded having worked as a teacher in a school or other educational institution. The analysis of the 15 responses in the *"Other"* category reveals that 9 teachers have education experience through work or volunteering activities (for example, *"Consulting workshops"*, *"Volunteered teaching coding at two different elementary schools."*, *"Yes, teaching assistant at my university, tutoring high schools students, exam trainings, code-/techcamps during summer, and 2 year CoderDojo."*)

Q0.5 concerns the programming languages that the teachers know well enough to program in. The majority of the respondents (90%) can program in at least one commercial programming language; the remaining responses included only languages used for education, like Scratch, Blockly and Basic. The most popular programming language among the respondents is Python (66% of the respondents), followed by JavaScript (48%), Java (43%), C (30%), C++ (25%), PHP (22%), and other languages.

Table 2: Summary of respondent's demographics and education profile

*[CS]: Software development / computer science / informatics , [Yes]: Yes, I have worked as a teacher in a school or other educational institution.

Gender		Age		Degree		Educ. Exper.	
Female	48	<24	4	[CS]*	**48**	[Yes]*	36
Male	**50**	24 - 30	14	Teaching/Educ.	18	No	**47**
		31 - 38	25	Other	32	Other	15
		39 - 46	**28**				
		47 - 53	15				
		>53	12				

Most of the code club teachers are between 31 and 46 years old, have a main degree related to computer science (49%) or education (18%), have no education experience (48%), and can program in at least one commercial programming language (90%). 10% of the code club teachers can only program in languages used for education.

4.2 RQ1: Code clubs operation

The operation of code clubs in terms of participants, lesson material and style and the use of assessments was explored through the data collected in Q1.1 to Q1.9. In Table 3 we summarize the data collected about the students of code clubs. We use the mean value and the five-number summary to describe the reported number, age and gender of the students.

Table 3: Summary statistics of data collected about the students of code clubs

	mean	min	Q1	median	Q3	max
Q1.1 Number of students in the code club	20,37	4	10	16	25	150
Q1.2 Age range: youngest student	8,53	4	7	8	9	16
Q1.2 Age range: oldest student	13,28	9	11	13	16	25
Q1.3 Percentage of female students	30%	0	20%	30%	40%	80%

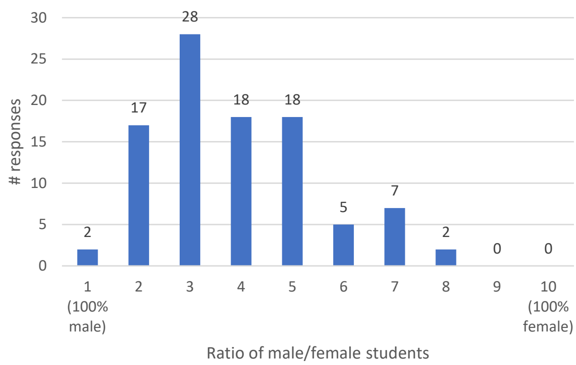

Figure 1: Responses to Q1.3 on a 10-scale ratio from 100% male to 100% female

Most code clubs have a small number of participants; half of the code clubs have up to 16 students, while 75% of them have up to 25 students. Three answers reported an unrealistically large number of students (over 50 and up to 150), even though this question included an explanation that *"If you teach at more than one, please reply to our questions about the one that you most frequently teach at"*.

The ages of the students vary greatly, from a code club that was reported to teach students from 4 to 10 years old to another whose students are from 16 to 25. The students of most (75%) code clubs, however, are from up to 9 to up to 16 years old. The average minimum age of students is 8,53, while the average maximum age is 13,28.

Female students are under-represented at code clubs; the average percentage of female students (out of 97 responses) is 30%. Figure 1 plots the distribution of the teacher's responses. Only 14 out of 97 code club teachers reported more than half of their students being female, while even two teachers answered that they have 100% male students.

Most of the teachers replied that their code club is part of a program, specifically CoderDojo (36%), Code Club (31%) or country-specific programs (11%). The remaining code clubs (22%) are not part of any program. Examining the student profile, the students attending Code Club classes are younger (average range from 8 to 11 years old) and are more often female (33%) than in CoderDojo clubs (26%). A larger variety of languages are taught at CoderDojos (average 2,9, median 3) than at Code Clubs (average 2,3, median 2). The majority of the code clubs have weekly (58%) or monthly (32%) sessions. The rest of the code clubs operate either every two weeks (5 responses), yearly (2) or randomly (3).

The most commonly taught language is Scratch (in 87 of the 98 responses, or 89%), followed by Python in almost half (48%) of the code clubs, followed by Arduino in 17 code clubs, Mindstorms in 16, Micro:bit in 15, HTML in 13, Java in 11, JavaScript in 10, Blockly in 4, C-like languages (e.g. ArduC, RoboC or NXT-C) in 3, and other languages. Within the answers in the *"Other"* field we also found Sonic Pi, Blender, Snap!, Swift, MBlock, Spheros, Flowol4, Crumble, Codebug, Node JS, Lightbot and A.l.e.x. Differences were found between the languages taught in code clubs of different programs. For example, 7 out of the 11 code clubs that teach Java are CoderDojo clubs. Certain languages, for example JavaScript, are taught mainly by people with computer science degrees (8 out of the 10 responses), while others, for example Java, are taught mainly by male teachers (9 out of 11 responses).

Answering whether they use a lesson plan, most of the teachers (45%) selected *"I do not use a lesson plan"*. Some declared using Code.org lessons (13%) or the Scratch Creative Computing Handbook (5%). In the open-text responses, 16 teachers described using their own lesson plan answering, for example, *"I plan it myself drawing on multiple sources for inspiration"* or *"I have my own lesson plan. Which is quiet flexible to change based on the interest and ability of my students."*. 11 teachers declared using Code Club material and 6 CoderDojo material. Other answers on lesson plans included *"lessons for Microbit"*, *"Code combat"* and *"Swift Playgrounds"*.

Regarding the style of teaching at the code club, the majority of teachers (70 out of 98, or 71%) selected that *"The students work independently on their own projects, and I help if they need me."* In 33% of the code clubs the teachers also give plenary sessions addressing the students. A teacher further explained that *"I occasionally do a session where I will introduce something different, such as Physical Computing."*. In 37% of the code clubs the students work in groups and they help each other where needed. Multiple styles are often combined, as a teacher explains: *"Project-based blended learning including self-learning, facilitator-led sessions, pair-programming"*. Several open-text responses also give insight to the degree of instruction that the students receive, for example *"I give them an assignment and then let them work in pairs independently (answering question where they have them)"* or *"they get a game with the code on paper and rebuild the game."*

In terms of assessment, the most common form is reported to be giving out stickers or badges for achievements (29 out of 62 responses, or 47%). Stickers might not always used for achievements, though; a teacher added that *"[...] I usually have stickers to hand out at the end of the day. But they get them just for participating."* Other used forms of assessment are quizzes or tests to get feedback on student understanding (18%), and programming competitions (16%). The use of formal tests for grading the students is stated

Table 4: Difficulties in learning to program as perceived by code club teachers (98 responses)

Difficulty	Frequency
Debugging and error messages	55 (56,12%)
The way of thinking is unfamiliar to them	45 (45,92%)
Abstract thinking	45 (45,92%)
The syntax of programming languages	30 (30,61%)
Concepts related to variables	26 (26,53%)
Concepts related to functions	23 (23,47%)
Concepts related to loops	19 (19,39%)
Concepts related to conditions	15 (15,31%)
Commitment or engagement with the lessons	12 (12,24%)
Motivation and interest	10 (10,20%)

in only 3 responses (5%). This question received a large amount of open-text responses. Most of them (11 out of 25) report using no form of assessment. Some answers describe informal or verbal assessment of student understanding, for example *"Encourage them to show me what they have created"*, or *"I talk with my pupils about their learning process. Puzzles get solved so I can track progress and understanding"*. We found answers reporting self-assessment of the students, for example *"Students self-rank their understanding of a subject on a 1 - 3 scale before we move to a new subject"*, as well as peer assessment, for example *"sharing work at the end and peer assessment"*, or *"only presentation but they are more aimed at presenting skills and inspiring each other"*.

> Most code clubs have a small number of mostly (70%) male students of varying ages. They are usually part of the CoderDojo, Code Club, or other national programs. They teach Scratch (89%), Python (48%), and several other languages. The students work independently on their own projects (71%) and, in some code clubs, the teachers give plenary sessions (32%). Formal assessments and grading are rare; more common are stickers or badges for achievements (47%) and other forms of formative assessment.

4.3 RQ2: Learning barriers

All 98 teachers replied to question Q2.1 and identified what the students struggle with at their code club by selecting at least one out of the 10 alternative options or by contributing their own response. Teachers identified an average of 3,13 (and a median of 3) difficulties faced by their students, as shown in Table 4.

The majority of the teachers (56%) answered with debugging and error messages in what their students struggle with. A teacher described this further as *"dealing with the frustration of inexplicable or unwanted behaviour of a program, and finding ways to solve it."* Almost half of the teachers (46%) identified the unfamiliar way of thinking and abstract thinking as a difficulty. A teacher whose students are from 8 to 12 years old related the age of the students to that, reporting that (translated from Dutch) *"sometimes children that register for my workshops are too young and find abstract thinking too difficult to really understand what they are doing."*. The syntax of programming languages was identified as a barrier by less than a third (31%) of the teachers.

The least popular answers related to the commitment (identified as a learning barrier by 12 teachers, 10 of which were male) and the motivation of the students (identified by 10 teachers, none of which were teaching at Code Clubs).

Of the programming concepts that we explored, the ones most often selected as the ones that students struggle with are variables (27%) and functions (23%). Loops are reported as difficult by 19 teachers, the majority of which (13 out of 19) are also teaching text-based languages (Python, JavaScript, Java and HTML/CSS) at their code clubs.

This question received 25 open-text responses, 20 of which were categorised as seven new types of difficulties. Those new types, ordered by frequency of appearance, are: language-specific difficulties, creativity, child-dependant (age, experience) difficulties, concentration problems, computer interaction (e.g. typing) issues, fear of making mistakes and focus on language. Language-related difficulties were identified for Scratch coordinates, lists and the project management interface. Teachers identified struggles related to creativity, for example *" thinking for themselves instead of blindly following the tutorial"* and *"design something for themselves and implement that."* Related to concentration, a teacher reported that students get *"distracted by playing games."* Another teacher reported the student focus on language as a learning barrier, as students *"often become focussed on learning Scratch itself, rather than building higher-order skills."*

> Teachers of code clubs most commonly identify debugging/error messages and abstract thinking as the difficulties that their students face in their code clubs. Commitment and motivation are rarely reported as barriers. Variables and functions are the concepts most frequently reported as difficult.

4.4 RQ3: Perceived gender differences

Question Q3.1 of the survey requests teachers to identify the ways in which boys are different from girls in their programming classes. It contains six sub-questions, shown in Table 5, for which teachers had to select from a five-scale range from Boys to Girls. We collected 96 replies for each of these sub-questions.

In Table 5, as well as in Figure 2, it can be seen that teachers identify gender differences especially for two traits, namely confidence and concentration. In the question *"Who is more confident?"*, 48 out of the 96 responses (50%) lean towards boys. Open-text responses in Q3.2 provide more insight into this. Several teachers describe reduced confidence or self-efficacy for girls in their classes, for example *"I get initial "I will never understand this" reactions way more from girls than from boys. Completely invalidated after an hour or so of course, but still saddens me"*, or *"Boys overestimate themselves and girls underestimate themselves."*

On the other hand, in the question *"Who seems to concentrate better?"* 62 out of the 96 responses (65%) lean towards girls and even 23 responses (24%) define girls with certainty. Open-text responses of teachers explain that *"Girls tend to stay on-task more, whereas some boys can be easily distracted"* and *"Girls most often seem more eager and have a longer attention span which helps them think and work on a problem longer and more thoroughly"*. Persistence is also a trait that is more often attributed to girls, with the responses to the

Table 5: Frequency of responses to question Q3.1 on gender differences

Subquestion	Boys	Neutral, but maybe boys	Neutral	Neutral, but maybe girls	Girls
Who is more confident?	18	30	38	9	1
Who is more motivated to learn programming?	10	19	42	20	5
Who seems to like programming more?	8	18	60	9	1
Who is more persistent when something does not work?	2	12	47	23	12
Who seems to get it more easily?	4	11	62	13	5
Who seems to concentrate better?	0	4	30	39	23

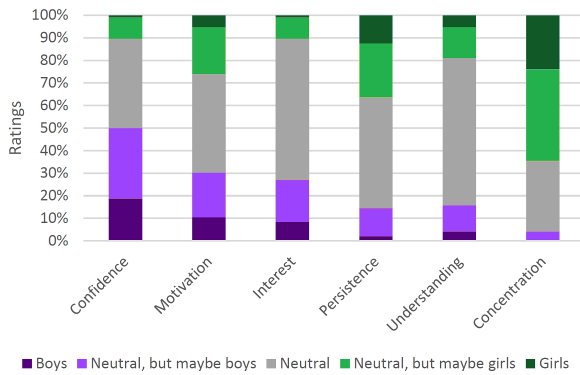

Figure 2: Percentages of responses on gender differences as perceived by code club teachers (96 responses)

question *"Who is more persistent when something does not work?"* leaning more towards girls (36%) than towards boys (15%).

For the motivation, interest and understanding traits, the responses were found to be balanced, with the majority of the responses being neutral between the genders. There was a single answer reporting perceptions of gender differences on innate skills, that *"algorithmic thinking seems to be more natural to boys"*. Contradictory textual responses relating to motivation and interest have been given. One teacher reported that *"Girls like to learn how to make "applications" (websites, games, etc.) Boys like to learn how to program. Both learn equally fast."*. Several responses, on the other hand, were in the lines of *"We have a cohort of boys that just play games"* and (translated from Dutch) *"The boys are more interested in testing games than programming themselves."*

We further examined individual correlations between the teachers' gender beliefs and their gender, age group, main degree and education experience, and found no significant effects.

The open-text Q3.2, on what other differences the teachers have noticed between how boys and girls learn programming, received 54 responses. The analysis of these responses resulted in the following additional types of gender differences (ordered by frequency of appearance):

Preferred type of projects, where girls are reported to prefer storytelling and visual/creative exercises, whereas boys prefer implementing games. Several answers are in the lines of *"The games they create are very different. Girls usually create games such are very colourful and vibrant, boys want to shoot things and focus less on the design"* and *"Girls seem to tend to like the more creative assignments, like computationally generated drawings whereas the boys tend to be more interested in the robots/less 'drawing'-like exercises."*

Didactic preference and responsiveness to instruction is increased for girls. For example, *"Girls seem to prefer structured sessions"*, *"Girls are happy to follow the instruction to the letter, boys want to add their own spin"*, and *"Boys just start blindly without reading lessons and then run into trouble pretty quickly, then call for help. Girls tend to focus more, start reading and ask questions when they're really stuck."*

Collaboration skills are described to be increased for girls, as *"Girls tend to talk and discuss more when working in partnerships whereas boys tend to have one who takes the lead"*, and *"Girls are more likely to help each other before asking me for help. Boys tend to ask for help from me first. Boys seem to work more independently where girls like group work."*

Grit and focus are often attributed to girls with responses like *"Girls are a bit apprehensive but when they get started they go go go! They make progress very easily and are more determined. Girls have more grit to finish the puzzle and are more precise"* and *"There are some boys who really focus and can go step by step through the projects, whereas some boys rush through without really understanding what they are doing. [...] All of the girls in my club have always been more careful and* *methodical. They seem to want to understand what they are doing more and don't mind taking their time.".*

Structuredness in the way of approaching problems, where *"Boys get straight in with their coding with quite elaborate ideas. Where as girls start simpler and focus on planning."*

Preferred programming environment , where *"Girls in our club seem to stick to scratch, boys go for Micro:bit"*, or *"I have noticed that boys are more attracted to hardware than to software."*

Familiarity and prior knowledge is reported to be reduced for girls. For example, *"Boys tend to have more experience/ knowledge, so they meet fewer problems"* and *"Initially, girls are more hesitant to experiment and play around with the computers, and more scared of 'spoiling' them. While technology literacy levels are same, technology agency levels higher in boys initially."*

Apart from gender differences, answers to the open-text question have also revealed other differentiating factors. Among these, we note the effect of parental influence, with a teacher answering that *"The biggest difference I've seen is kids who have parents who code vs those who do not. The gender is less of a factor than I was expecting."* and the effects of culture or stereotypes: *"Main issue is when they get older, the culture tells girls it's not cool to do coding."*

Teachers most commonly identify the boys in their code clubs as being more confident and the girls as being more persistent and able to concentrate better. Some teachers indicate that boys get distracted from playing games instead of learning to program. Gender differences are commonly reported in the preferred type of projects, the responsiveness to instruction, collaboration skills, grit and focus, and structuredness, which are considered increased for girls, as well as familiarity and prior knowledge, which are increased for boys.

5 DISCUSSION

Our study has highlighted several differences between code club lessons and in-school programming classes. Around half of the code club teachers that replied to our survey have a computer science-related background and no education experience, while students are perceived as motivated and committed. Moreover, the classes are small, the taught material varies greatly, students work mostly independently, and assessments are formative and informal.

There are also differences between different programs. CoderDojo teachers reported teaching a larger variety of programming languages, while Code Club teachers reported having students of younger ages and a higher ratio of female students in their clubs. This is also evident in the results of the annual surveys where, for 2018, Code Club reports on having 40% female attendees, while CoderDojo 33%, with 65% of the CoderDojo teachers having less than or around 25% female students [7, 8]. In terms of teachers, while in our survey the respondents of Code Club and CoderDojo clubs have similar distributions in their main degree and education experience, the annual surveys reveal that the number of professional educators is different for Code Club (55%) than for CoderDojo (25%). It is also recognized that having female educators who can act as mentors and role models might lead to a higher proportion of female participants in the code clubs [8].

Learning barriers. Our findings related to learning barriers perceived by code club teachers are different from the findings of Dorn *et al.* for secondary school teachers [9]. Both studies identify programming language/ syntax and abstract thinking/abstraction capacity as a common difficulty. However, we found that code club teachers report debugging and error messages as the most common difficulty that their students struggle with, which was also reported in the study by Dorn *et al.*, but not commonly. At the same time Dorn *et al.* found that commitment and motivation/interest are commonly reported as difficulties for students to learn programming, unlike in our study, where commitment and motivation were the most rarely reported barriers. We believe that these differences can be attributed to the self-selection of the code club courses by the students, which makes them more likely than school students to be self-motivated to learn. Butler *et al.* have identified the motivating factors of CoderDojo participants as provision of support, scope for creativity, freedom of choice, provision of challenge, and friendships, and have even reported on some students describing attending CoderDojo in terms of integrated motivation, believing it to be consistent with their values and needs [4].

The programming concepts that were found to be the most commonly reported as difficult for the students were variables and functions, followed by loops and conditions. Seiter and Foreman [21] have found that, for elementary school students, conditionals and variables were under-represented in the projects of students before grade 5. In a large dataset of public Scratch programs it has also been found that programming abstractions related to variables and functions were underutilized [1]. However, loops were frequently found in the dataset. On closer inspection of our data, we found that the majority (13 out of 19) teachers that reported loops as a difficulty were also teaching text-based languages at their code clubs. This finding might therefore relate to the differences in loop-related syntax and semantics between block and text based languages.

Gender differences. The survey revealed several types of perceived gender differences among students, with the most commonly reported ones being confidence, persistence and concentration. Differences perceived by school teachers in the confidence, structuredness and collaboration of their students have also been reported in the study of Funke *et al.* [11]. Unlike their study work, however, we have not received any answers related to differences in scientific curiosity, creativity or frustration. On top of the differences that they have identified in their work, our survey revealed differences in concentration, motivation, the preferred type of projects, didactic preferences and grit and focus. The differences between the two studies could, however, be attributed to discrepancies in the categorization of the answers.

The most commonly reported gender difference in the open-text question of our survey was the preferred type of projects, where girls were reported to prefer storytelling and visual projects instead of games. This is in line with findings in [12], where it was found that boys used more motion-category blocks and girls used more looks-category blocks in their Scratch programs. In the open-text question, collaboration skills and practices were also commonly identified differences by the teachers, which is in accordance with the findings of Chetty and Barlow-Jones, where girls reported more than boys to benefit from communication and collaboration [5].

Gender differences related to confidence could have several implications. In prior work we have found that, for female elementary school students, self-efficacy is strongly correlated with how attractive they view computer science as a career path [2]. Since self-efficacy or confidence differences can potentially help explain the gender participation gap in computer science, we believe that they are worth examining further.

5.1 Threats to validity

A threat to the external validity of our study concerns the code club teachers that answered our survey, who might not be representative of all people who teach at code clubs. In order to collect answers from a large a diverse set of teachers, we invited them through several communication channels with global outreach for a total of six months. To ensure that only people who teach at after-school programming clubs answer the survey, we clarified it in the invitation messages and in the introduction of the survey. Moreover, the survey started with the questions that are most specific to code clubs, namely with Q1.1 *"How many children are in your code club?"*, followed by the rest of the questions in the first group.

With respect to construct validity, respondents could possibly misinterpret some questions. For example, in Q1.6 *"Is your code club part of a program?"*, the word program could be interpreted in various ways. To make our intentions clear to the respondents, they were offered choices like *"CoderDojo"*, *"Code Club"*, etc. We also designed the survey to use clear language, avoiding specialized terms. For example, Q1.9 *"Do you use any form of assessment?"* offered alternative choices that did not include the terms "formative" or "summative", which people without an education background might not be familiar with. Finally, the survey was reviewed by eight colleagues who reported their observations about possible ambiguities.

A threat to the internal validity of our study is the social pressure that the respondents might have felt when disclosing opinions about gender-related issues. While self-administered surveys apply less social pressure than interviews, participants might have been inclined to respond with *"Neutral"* to all questions in Table 5. To support the respondents in expressing their opinions, we included the intermediate scales (e.g.,*"Neutral, but maybe boys"*), as well as open question Q3.2, which is expressed in a way that invites them to explore which other differences they have noticed.

6 CONCLUDING REMARKS

The goal of this paper is to examine how programming is taught at code clubs and to explore the experiences and perceptions of code club teachers. We investigated the learning barriers that the teachers find their students facing during the programming lessons, as well as the gender differences that they identify among their students.

Our study highlights several characteristics of code club lessons the differentiate them from in-school programming classes, including the teacher background being often not related to education, the diversity in the teaching material, the teaching style which includes mostly independent work, and the absence of summative assessments. The differences were also reflected in the learning barriers perceived by the teachers, with commitment and motivation being rarely reported.

Several gender differences are identified by the teachers, including differences in confidence, persistence, concentration, collaboration skills, and even preferred types of projects. We should highlight, however, that these differences pertain teacher perceptions, which could be biased by general gender perceptions or stereotypes. In future work, we plan to work towards examining the validity of these perceptions though studies that involve observations and, possibly, interviews with code club students.

ACKNOWLEDGMENTS

The authors would like to thank Krishna Thiruvengadam for his valuable help in publicising the survey. We would also like to thank the code club instructors that were involved in this research. We are grateful for your time and for sharing your experiences and insights with us.

REFERENCES

[1] Efthimia Aivaloglou and Felienne Hermans. 2016. How Kids Code and How We Know: An Exploratory Study on the Scratch Repository. In *Proceedings of the 2016 ACM Conference on International Computing Education Research*. ACM, 53–61.

[2] Efthimia Aivaloglou and Felienne Hermans. 2019. Early Programming Education and Career Orientation: The Effects of Gender, Self-Efficacy, Motivation and Stereotypes. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. ACM, New York, NY, USA, 679–685. https://doi.org/10.1145/3287324.3287358

[3] Rosangela Bando, Emma Naslund-Hadley, and Paul Gertler. 2019. *Effect of Inquiry and Problem Based Pedagogy on Learning: Evidence from 10 Field Experiments in Four Countries.* Working Paper 26280. National Bureau of Economic Research. https://doi.org/10.3386/w26280

[4] Natalie Butler, Cliona Flood, and Andrew Power. 2018. What Motivates a Ninja? An exploration of students' CoderDojo experience. In *Cyberpsychology and Society: Current Perspectives*, Andrew Power (Ed.). Routledge.

[5] Jacqui Chetty and Glenda Barlow-Jones. 2018. Coding for girls: dismissing the boys club myth. In *the 18th International Conference on Information, Communication Technologies in Education (ICICTE 2018)*.

[6] Jody Clarke-Midura, Frederick Poole, Katarina Pantic, Megan Hamilton, Chongning Sun, and Vicki Allan. 2018. How Near Peer Mentoring Affects Middle School Mentees. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. ACM, New York, NY, USA, 664–669. https://doi.org/10.1145/3159450.3159525

[7] Jonathan Dickins. 2018. *Code Club Annual Survey 2018 Report.* Technical Report. Raspberry Pi Foundation. https://www.raspberrypi.org/app/uploads/2019/04/CoderDojo-Annual-Survey-2018.pdf

[8] Jonathan Dickins. 2018. *CoderDojo Annual Survey 2018 Report.* Technical Report. Raspberry Pi Foundation. https://www.raspberrypi.org/app/uploads/2018/05/Code-Club-Annual-Surveys-2017.pdf

[9] Norbert Dorn, Marc Berges, Dino Capovilla, and Peter Hubwieser. 2018. Talking at Cross Purposes: Perceived Learning Barriers by Students and Teachers in Programming Education. In *Proceedings of the 13th Workshop in Primary and Secondary Computing Education (WiPSCE '18)*. ACM, New York, NY, USA, Article 12, 4 pages. https://doi.org/10.1145/3265757.3265769

[10] Deborah A. Fields, Lisa Quirke, Janell Amely, and Jason Maughan. 2016. Combining Big Data and Thick Data Analyses for Understanding Youth Learning Trajectories in a Summer Coding Camp. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*. ACM, New York, NY, USA, 150–155. https://doi.org/10.1145/2839509.2844631

[11] Alexandra Funke, Marc Berges, Andreas Mühling, and Peter Hubwieser. 2015. Gender Differences in Programming: Research Results and Teachers' Perception. In *Proceedings of the 15th Koli Calling Conference on Computing Education Research (Koli Calling '15)*. ACM, New York, NY, USA, 161–162. https://doi.org/10.1145/2828959.2828982

[12] Alexandra Funke and Katharina Geldreich. 2017. Gender Differences in Scratch Programs of Primary School Children. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education (WiPSCE '17)*. ACM, New York, NY, USA, 57–64. https://doi.org/10.1145/3137065.3137067

[13] Shuchi Grover and Roy Pea. 2013. Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher* 42, 1 (2013), 38–43. https://doi.org/

10.3102/0013189X12463051 arXiv:https://doi.org/10.3102/0013189X12463051

[14] Felienne Hermans and Efthimia Aivaloglou. 2017. Teaching Software Engineering Principles to K-12 Students: A MOOC on Scratch. In *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track*. IEEE Press, Piscataway, NJ, USA, 13–22.

[15] Jon A. Krosnick and Stanley Presser. 2010. *Handbook of Survey Research* (2nd ed.). Emerald, Chapter Question and Questionnaire Design, 263–314.

[16] James Lockwood and Aidan Mooney. 2017. Computational Thinking in Education: Where does it fit? A systematic literary review. *arXiv preprint, arXiv:1703.07659* (2017).

[17] John H. Maloney, Kylie Peppler, Yasmin Kafai, Mitchel Resnick, and Natalie Rusk. 2008. Programming by Choice: Urban Youth Learning Programming with Scratch. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*. ACM, 367–371.

[18] Nigel McKelvey and Pamela Cowan. 2017. Valence at CoderDojos: An Exploration. *Literacy Information and Computer Education Journal (LICEJ)* 8, 1 (March 2017), 2525–2533.

[19] Orni Meerbaum-Salant, Michal Armoni, and Mordechai (Moti) Ben-Ari. 2010. Learning Computer Science Concepts with Scratch. In *Proceedings of the Sixth International Workshop on Computing Education Research*. ACM, 69–76.

[20] Jenny Rowley. 2014. Designing and using research questionnaires. *Management Research Review* 37, 3 (2014), 308–330. https://doi.org/10.1108/MRR-02-2013-0027 arXiv:https://doi.org/10.1108/MRR-02-2013-0027

[21] Linda Seiter and Brendan Foreman. 2013. Modeling the Learning Progressions of Computational Thinking of Primary Grade Students. In *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research (ICER '13)*. ACM, 59–66.

[22] Irene Sheridan, Deirdre Goggin, and Linda O'Sullivan. 2016. Exploration of Learning Gained through CoderDojo Coding Activities. In *INTED 10th International Technology, Education and Development Conference*.

[23] Neil Smith, Clare Sutcliffe, and Linda Sandvik. 2014. Code Club: Bringing Programming to UK Primary Schools Through Scratch. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14)*. ACM, New York, NY, USA, 517–522. https://doi.org/10.1145/2538862.2538919

[24] Jennifer Tsan, Kristy Elizabeth Boyer, and Collin F. Lynch. 2016. How Early Does the CS Gender Gap Emerge?: A Study of Collaborative Problem Solving in 5th Grade Computer Science. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*. ACM, New York, NY, USA, 388–393. https://doi.org/10.1145/2839509.2844605