

Engineering Students' Experiences of Assessment in Introductory Computer Science Courses

Emma Riese¹ and Stefan Stenbom²

Abstract—Contribution: This study evaluates the generalizability of previously identified perceptions among engineering students of assessments in introductory programming (CS1). The students' perceptions of their instructors' and teaching assistants' (TAs) roles in these assessments are also studied, and differences based on prior programming experience, gender, and course explored.

Background: Basic programming skills are desirable also for students who are not majoring in computer science (CS). Students' experience of assessments has not been fully explored.

Research Questions: 1) How do engineering students experience the assessment (lab assignments, midterm exam, and project) in their CS1 courses? 2) What are the students' perceptions of the TAs and instructors in relation to these assessment situations? 3) What significant differences can be found based on students' prior experience in programming, gender, and course?

Methodology: Previously identified themes from an interview study worked as a framework for the formulation of 25 statements used in a survey among 137 students in six CS1 courses (second part of a mixed-method study). Descriptive statistics, Mann–Whitney U Test, and Kruskal–Wallis tests with Bonferroni corrections, were used to analyze the data.

Findings: Laboratory assignments were experienced as an opportunity to learn while the exams were viewed as predictable. The projects were perceived as authentic, although varying in difficulty, and as a huge leap from the lab assignments. Students perceived the instructors to put their touch on the course but viewed their TAs as carrying out the assessments, and experienced variations between TAs. Female students experienced these variations to a larger extent and perceived received feedback as less useful.

Index Terms—Assessment, first year, higher education, students experience.

I. INTRODUCTION

INTRODUCTORY programming, the first computer science (CS) course (CS1) that many students enroll in during their first year in college or at university, has been the focus of many research studies [1]. However, it has also been found that students face lots of challenges when learning how to program [2]. That programming is experienced as difficult to learn is not only attributed to the content [2], but also by social factors, such as a defensive and competitive classroom

Manuscript received 11 February 2022; revised 28 October 2022 and 20 December 2022; accepted 23 December 2022. Date of publication 3 February 2023; date of current version 4 August 2023. (Corresponding author: Emma Riese.)

Emma Riese is with the Department of Computer Science, KTH Royal Institute of Technology, 10044 Stockholm, Sweden (e-mail: riese@kth.se).

Stefan Stenbom is with the Department of Learning in Engineering Sciences, KTH Royal Institute of Technology, 10044 Stockholm, Sweden.

Digital Object Identifier 10.1109/TE.2023.3238895

climate [3]. CS1 courses have been found to have a bimodal grade distribution, which suggests that a subset of students are struggling while another subset pass through the courses with ease [4]. CS courses often have large enrollment as many education programs also for non-CS majors include introductory programming [5]. A quite common approach to ease the course coordinators' (main instructors') workload is to employ teaching assistants (TAs) [6]. The TAs are themselves students who, for example, can assist with conducting tutorials and grading assignments [6].

Although CS1 courses have been well researched, students' experiences of assessment situations have been identified as a gap in [1], which this study aims to fill. This study aims to identify how non-CS majors, engineering students, experience assessment situations in their introductory CS courses and how they perceive the course coordinators' and TAs' roles in these situations. This student group is particularly interesting to study since CS and programming are mandatory parts of many engineering education programs and considered useful knowledge and skills to master for students also with other majors [5]. Although these students have not chosen to study CS as their major and might not view themselves as future programmers or computer scientist, they make up a large part of the student body in CS1 courses. In the courses where the reported data collection took place, the courses are also tailored to fit within the students', respectively, education programs and offered specifically for non-CS majors. This article reports on the generalizability of the perceptions previously identified (in [7]) and explores the differences based on prior programming experience, gender, and course offering. The following research questions (RQs) are addressed.

- 1) How do engineering students experience the assessment (lab assignments, midterm exam, and project) in their CS1 courses?
- 2) What are the students' perceptions of the TAs and instructors in relation to these assessment situations?
- 3) What significant differences can be found based on students' prior experience in programming, gender, and course?

To address the RQs, an exploratory mixed-method approach was used. The first qualitative step [7], informed the second quantitative step, which is reported in this article.

II. BACKGROUND

In this section, the theoretical foundation for this research is presented together with key concepts, and related work.

A. Theoretical Perspectives

This study uses a social constructivism view of learning, commonly used in computing education [8]. Constructivism builds on the notion that knowledge is constructed by the learner based on their previous knowledge and experiences. Within social constructivism, learning is seen as a social phenomenon in which knowledge is constructed within a culture or group as a result of social interactions [8]. One of the key concepts in social constructivism is the *zone of proximal development* (ZPD). Vygotsky [9] defined ZPD as the space between what a learner can do alone and what the learner can do with the assistance of or in collaboration with more expert others. Within the ZPD, a more knowledgeable person can aid the learner to complete a task, too difficult to manage independently [10]. This theory has also been applied within CS education in previous studies where students' ZPD have been identified and adjusted [11], [12], [13]. Another key concept is constructive alignment [14]. Constructive alignment states that the intended learning outcomes should be aligned with the learning activities and the assessments. The aim with using constructive alignment for organizing the education is that learners can create meaning from the learning activities directed at fulfilling the intended learning outcomes, which is also what is assessed [14].

B. TAs in Computer Science Education

Employing TAs has been described as a possible solution to handle the increased enrollment within CS [15]. The TAs' responsibilities could, for instance, include giving tutorials, holding office hours, and grading assignments [6]. TAs are themselves students, often close in academic age to the students they teach, hence often described as more approachable than instructors [16]. However, TAs have also been found to not be properly trained for their duties, and particularly experiencing uncertainties and challenges with the assessment situations [17], [18]. In regards to grading, it has also been shown that grading that TAs have carried out in a group setting has higher reliability than grading done in a solo-setting [19].

C. Assessment Approaches

Assessment of students' achievements is a critical feature of formal education as it forms the documentation for degrees [20]. Furthermore, assessment plays a vital role in identifying students' ZPD and guiding students' further engagement with the learning activities [21]. Assessment is often divided into two different categories: 1) formative and 2) summative assessment [22]. The summative assessment aims to evaluate and grade the students' knowledge or competency at a given time. In contrast, formative assessment is a type of assessment that aims to direct and support the students learning by providing useful feedback. However, a single assessment situation could have both these aims [22].

Assessment in introductory CS is a research area that has grown during the last years [1]. Examples of such research include approaches to assessment and design of assessment situations [23], [24], [25], and handling of misconduct among students [26]. To provide the students with useful feedback,

has been proposed to improve learning in introductory CS [27]. One of the approaches to incorporate peer feedback is pair programming, a practice in which two students work together in interchangeable roles: one as the driver, the person writing the code on the keyboard, the other as the navigator, the person who is planning and pointing out mistakes [28].

One of the more researched summative assessment types in CS1 courses is exams that are conducted in computer lab settings [29]. There have also been studies comparing or combining pen-and-paper exams and computer-based exams [23], [30]. In CS1 courses, previous studies have explicitly focused on the use of multiple-choice questions and concluded that these types of questions could successfully be used to measure students' programming knowledge [31]. A disadvantage with pen-and-paper exams is that it is an inauthentic way of programming [30]. Authenticity has also been further explored from the student perspective [32] and shown to align well with faculty definitions [33]. There has also been an initiative to develop "real-world problems" assignments suitable for CS1 curricula [34]. Portfolio grading, where students build a portfolio that is graded at the end of the course, has also been tried as an alternative approach [24].

Students' experiences of assessments have been identified as a gap in CS1 research [1]. However, a previous study used a phenomenographic research approach to explore this without distinguishing between different assessment types [35]. The outcome space consisted of five categories, ordered in a hierarchy: 1) grading is important to the teacher; 2) grading is important to the student; 3) assessment as guidance; 4) assessment as an opportunity to learn; and 5) assessment as a way to communicate, where the top three categories are the most desirable [35].

D. Student Success Factors in Computing Education

Students' comfort levels, including being comfortable asking for help, have been shown to act as predicting factors for students' success [36]. A literature review on papers related to anxiety in CS education concluded that students' experiences of anxiety affect their learning and stems not only from the anxiety of learning how to program but also from other types of anxiety, such as test anxiety [37]. Collaborative learning (such as pair programming) has also been found to cause anxiety for some students [38]. It has also been found that the CS classroom can have a defensive climate, evaluative with a superiority, rather than a more desirable supporting climate [3]. A defensive climate is also viewed as particularly harmful for female students and other minority groups [3]. The stereotype of CS as a male subject has also been shown to be persistent and affect the sense of belonging for students [39]. Male students have also been found to start CS1 with more prior experience in CS than their female classmates, another predicting factor for success [40]. Female students have also been shown to, on average, receive lower grades in these courses [41]. Women have also been found to be underrepresented in CS education, in many countries and contexts, in numerous studies [42], [43], [44], [45], [46]. Common factors that lead to dropouts are lower confidence,

previous background, and sense of belonging in CS [41], [47], [48]. To address the fact that students enroll in CS1 with different prior experiences, some institutions have successfully divided their students into different cohorts based on prior experience [49]. A comprehensive review of recent studies on teaching and related practices that affect participation in computing identified several factors to enable broad participation and highlighted four main recommendations for instructions [46]. These were (in summary): avoid stereotypes in course material and teaching, emphasize and use collaborative learning practices, connect to students' lives and interests, and enable meaningful positive interactions with TAs and course coordinators [46].

III. METHOD

This section describes the research design, research setting, data collection, and method for data analysis.

A. Research Design

In this study, an exploratory sequential mixed-method research approach was used. This is a suitable approach when the RQs addressed are aiming to both explore phenomena and validate the generalizability of the findings [50]. The initial step is a qualitative data collection and analysis to explore the phenomenon. These results then act as a starting point for an informed quantitative data collection and analysis to investigate the generalizability. This article only reports on the results from this second step. In the first qualitative step [7], 11 engineering students, who had recently enrolled in different CS1 courses, were interviewed about their experiences and the data was analyzed by a thematic analysis [51]. The themes identified acted as a foundation for the statements used in this second quantitative step of the study, as seen in Table I. When developing the statements used in this quantitative step, the researchers also made sure to use the same type of language the students had.

B. Research Setting and Participants

Students from six different CS1 courses (C1–C6) were part of the data collection. Each course was taken by students who major in an engineering subject, or aim to do so but had not yet declared a specific major within engineering. All of the students enrolled in any of the courses were non-CS majors. See Table II for the course specifics. None of the courses requires any prior programming experience, and the students are not divided into the courses based on prior experience but on their educational programs, that is which major they have selected. Except for C4 and C5, the courses have different course coordinators. The course coordinator is the course's main instructor, the person responsible for planning and running the course.

The six courses followed a similar course design with similar learning objectives, typical for a CS1 course using the programming language Python. This includes data types, basic data structures, conditions, loops, functions, classes, and methods. In addition, C1 also had learning objectives about memory storage and version control. All courses had a very similar

TABLE I
THEMES IDENTIFIED IN THE QUALITATIVE STEP [7] AND MAPPING FROM THEMES TO STATEMENTS IN THE QUANTITATIVE STEP

Regarding	Themes	Statements
Lab assignments	As a necessary evil	S2-S3
	As a learning activity	S1
Exam	Predictable but not suitable	S6-S8
	Inauthentic assessment	S9
Individual project	A leap from the lab assignments	S10
	Learning by doing	S11
	Fair and authentic assignment and assessment	S12
	Difference in difficulty between projects	S13
Course coordinator	The designer of the course	S14
	The constructor of the exam	Excluded due to COVID-19 changes
	Somewhat absent	S15
TAs	Variation in quality depending on the TA	S17-S20
	Too few TAs	S25
	Helping hand?	S21-S23
	The TA as the person setting the grade	S16
Informal tutors and material	Ask your peers and older students	S4-S5
	Search the internet for answers	S24

TABLE II
OVERVIEW OF COURSES

Short name	C1	C2	C3	C4	C5	C6
Short description	Extra tasks, individual labs graded F–A	Assigned TA, labwork in pairs	Pair programming, assigned TA	Labwork in pairs	Labwork in pairs	Labwork in pairs
Study period (weeks)	20	20	25	20	20	30
Course credits (ECTS)	5	7.5	8	6	6	6
Number of lab assignments in Python	4	6	7	5	5	6
Nr. of students in the course	150	168	62	145	179	88
Nr. of students part of study - total(female)	27(11)	23(11)	16(4)	28(13)	25(7)	18(0)
No prior experience	10 (6)	9 (7)	7 (2)	14 (8)	13 (6)	6 (0)
Some prior experience	11 (5)	8 (4)	4 (0)	12 (5)	8 (1)	6 (0)
Lots of prior experience	5 (0)	6 (0)	5 (2)	2 (0)	4 (0)	6 (0)

structure regarding assessment situations with three assessment types: first lab assignments (4–7 assignments), then a midterm exam with multiple-choice/“fill in the gap” questions, and finally, a larger individual project. The lab assignments differed between the courses. An example of such an assignment is to construct a program that simulates a ticking clock by creating and using a clock class. The lab assignments were graded on a pass/fail scale in all courses except in C1. Students in C1 had to complete an extra task on each of the lab assignments to receive a higher grade than E (lowest passing grade), and they were also encouraged to work alone on the lab assignments. In the other courses, the students were encouraged to work in pairs, in C3, as a requirement after a pair programming introduction.

For the fall semester of 2020, the exam questions and the individual project descriptions were all taken from the same pool. During the midterm exam, each student took an online quiz, given at a predetermined time, with an individual set

of randomly selected questions. This setup was new and tried due to the COVID-19 restriction. The midterm exams were graded pass/fail (typically 80% needed to pass the exam) in all courses. For the projects, the students were allowed to choose which project they liked to work with from a large pool of projects. All projects consisted of different levels, corresponding to different grades. An example is to simulate a “memory” game, where the easiest level only needed to be text-based and handle three-letter words (E lowest passing grade, and D), while the advanced levels need to handle bad user input (C), different long words (B), and implement a graphic interface (A).

The TAs are not formally responsible for the grading but carry out the assessments and inform the person who is formally responsible (the person appointed as the examiner, typically the same person as the course coordinator). Since the beginning of 2020, all TAs have to enroll in a mandatory training course (total workload of 6 h) where assessment training is part of the content [52]. In C2 and C3, the students were assigned to a group (12–20 students), with an assigned TA, but in the other courses, the students did not have an assigned TA.

C. Data Collection

The invitation to participate in the survey was sent out through email to all students registered in any of the courses C1–C6. The surveys were distributed around a month after the last scheduled course activity. Three reminders followed the initial mailings. The data were collected anonymously, and the surveys were distributed in Swedish, the language of the courses. Students were asked to answer three (optional) questions about themselves regarding prior programming experience (no prior experience/some prior experience/lots of prior experience), gender (legal gender: female/male/chose not to answer), and course grade (F–A), respectively. In addition, data on which course the students had enrolled in were stored. It was entirely voluntary for the students to participate and answer any questions. No incentives to boost participation were offered. The students were informed of the purpose of the study, and informed consent for participation was collected through a final question of the survey. Since no interventions were made, nor was any sensitive data collected, no ethical approval was necessary according to the national and institutional guidelines on ethics where the study took place.

Altogether, the survey consisted of 25 statements (S1–S25), see Fig. 1 for the translated version of each of the statements. The statements were based on the previously identified themes [7] as shown in Table I. For each statement, the students were asked to rate how well they agreed with the statement on a 7-point Likert scale, ranging from 1 = *completely disagree with the statement*, to 7 = *completely agree with the statement*, with 4 being neutral.

D. Analysis

To address the first two RQs, descriptive statistics were used, and a chart (using Microsoft Excel) was drawn to visualize the data. To address the third RQ, nonparametric tests were conducted where the students were divided into groups

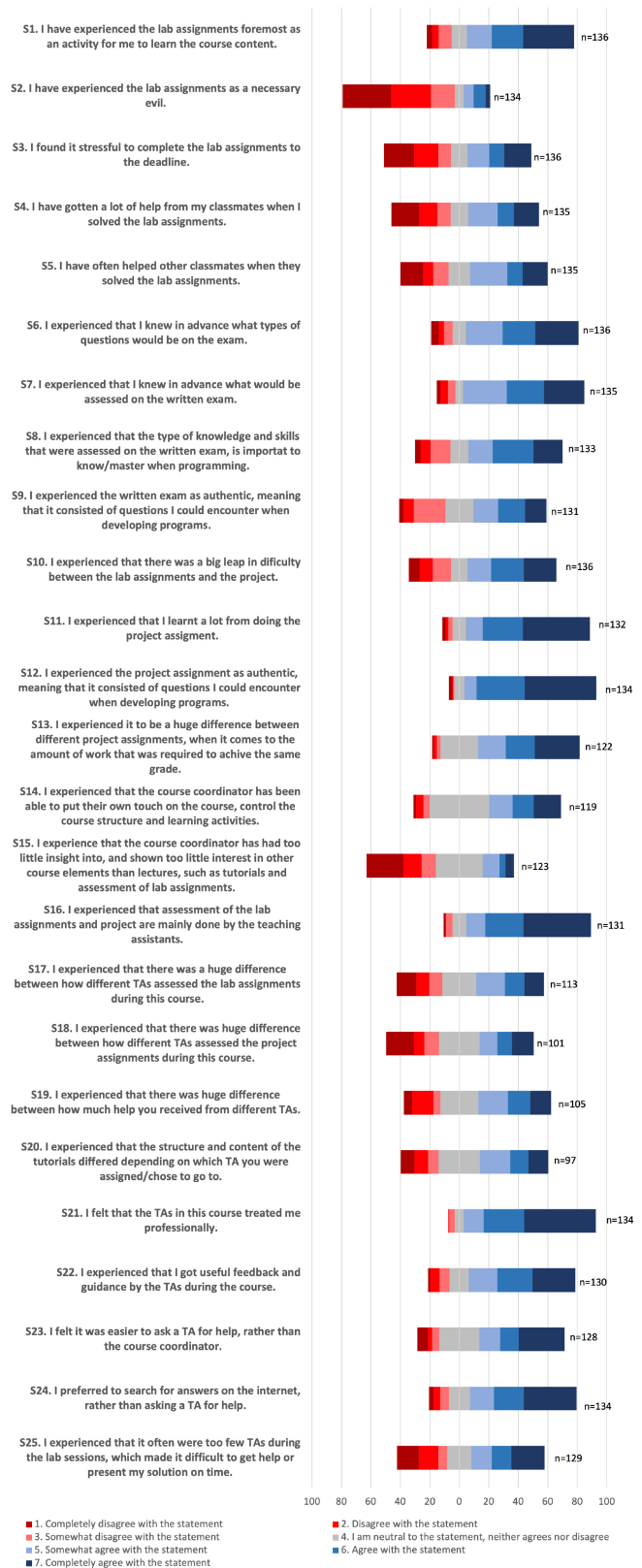


Fig. 1. Shows the likert scale data (in percentages) for each of the statements.

based on course enrollment, self-reported prior program experience, and gender, respectively. The hypotheses tested were that there would be no difference between the subgroups. To compare the answers based on gender, Mann–Whitney U tests

were conducted. A Mann–Whitney U test is a nonparametric test that can be used to compare the distribution in two independent groups based on ranks and is well-suited for ordinal data [53]. The two respondents who answered “choose not to answer,” were excluded. To investigate differences depending on course and prior experience, the Kruskal–Wallis test were used. The Kruskal–Wallis test is a nonparametric test comparing the distribution in independent samples consisting of more than two groups [53]. Where significant differences (at a 95% confidence level) were found, pairwise comparisons (Mann–Whitney U tests) with Bonferroni corrections were conducted to provide insights into how the groups differed. To investigate grade differences, the same type of nonparametric tests were used, but the grade point average (GPA) was also calculated. Letter grades are typically viewed as ordinal data, but are often treated as continuous data to calculate GPA [53]. All statistical tests were carried out using the software SPSS.

IV. RESULTS

This section presents the results from the survey data collection and analysis, structured around the three RQs.

A. Perceptions of Different Assessment Types

In Fig. 1, the survey results for the statements regarding the lab assignments (S1–S5), the exam (S6–S9), and the projects (S10–S13) are presented. In general, it was found that the students perceived the lab assignments as a learning activity and not as a necessary evil. Regarding stress, helping, and receiving help from classmates, there was a large spread in the answers and no clear trends. Regarding the exam, the students experienced knowing in advance what type of knowledge would be assessed, what type of questions would be asked, and viewed the skills and knowledge assessed as important. Many, but not all, students also viewed the exam as authentic. However, the projects were experienced as authentic by a larger percentage of the students. The students also experienced to have learned a lot from their projects, but that there was a huge difference between projects regarding the amount of work required to achieve the same grade. Most of the students also experienced a big leap in difficulty between the lab assignments and the projects.

B. Perceptions of the Course Coordinators and TAs

Regarding the course coordinator (Fig. 1, S14 and S15), the students experienced that they could put their touch on their course and did not perceive the course coordinator as having too little insights or interest in course activities that were not lectures. For the TAs (Fig. 1, S16–S25), the students perceived them to assess the lab assignments and projects. They felt they had been treated professionally by the TAs, received useful feedback and guidance, and that it was easier to ask a TA for help than their course coordinator. However, many of the students did state to prefer to search for answers on the Internet rather than asking a TA for help. The students' responses varied when it comes to differences between TAs (assessment of lab assignments, respectively, projects, how much help received, and content of tutorials). On a general

level, the respondents slightly agreed that there had been variations. Students also experienced that there could be too few TAs present, but the answers varied.

C. Differences

In Table III, the results from the statistical test regarding differences based on course enrollment, prior programming experience, and gender are presented. As could also be viewed in Table II, the female students participating in this study had less experience than the male students (Mann–Whitney U test resulted in a p -value: < 0.001). The female students also received significantly lower grades (Mann–Whitney U test p -value: 0.017). With the scale Not Finished (F) = 0, E = 1, D = 2, C = 3, B = 4, and A = 5, the GPA for male students was 3.60 with standard deviation (SD) = 1.842, for female students GPA = 2.96, SD = 1.821. Significant differences were also found between the students' grades when grouped on prior experience (Kruskal–Wallis test p -value: < 0.001). Pairwise comparisons with Bonferroni corrections showed students with no prior programming experience significantly differed from students with some experience (adj. p -value: 0.003) and lots of experience (adj. p -value: < 0.001). Students with some and lots of experience did not significantly differ (adj. p -value: 0.078). Students with no prior experience had a GPA = 2.60 and SD = 1.892, students with some experience had a GPA = 3.80 and SD = 1.568, and students with lots of experienced had a GPA = 4.46 and SD = 1.347.

V. DISCUSSION AND CONCLUSIONS

This section analyzes the results using the theoretical perspectives and the three RQs. Following that, is a discussion of limitations and threats to validity, and concluding the section is a discussion on implications for practice.

A. Findings in Relation to Theoretical Perspectives

To form meaning in the CS1 courses studied, this article applies the social constructivist approach to learning. As a result, learning is viewed as an active process where humans construct meaning in response to ongoing interactions with their surroundings. Teachers and TAs are responsible for designing and organizing learning and assessment activities, facilitating students' learning, and providing instructions within the students' ZPDs [9], [10], [14].

The findings of this study indicate that students utilize social constructivism, as affirmative answers to the asked statements predominate. Students experienced the assessment activities as formative learning activities for exploring the material further (see S1, S2, S9, and S11). They also emphasize the social dimension as they collaborate to progress (see for instance S4 and S5). Especially interesting are the differences in answers based on students' experience and gender. This relates to TAs and instructors' capacity to design learning and assessment activities and facilitate students' learning for all students. As outlined in the findings, there is a large variation based in prior programming experience, and male and female students', on several of the study aspects. To meet students' needs, instructors and TAs must develop strategies for identifying students'

TABLE III
RESULTS FROM THE KRUSKAL–WALLIS TESTS COMPARING SUBGROUPS BASED ON COURSE ENROLLMENT, RESPECTIVELY, SELF-REPORTED PRIOR PROGRAMMING EXPERIENCE, AND THE MANN–WHITNEY U TEST COMPARING SUBGROUPS BASED ON GENDER

	Statement	p-value course (C)	p-value prior experience (P)	p-value gender (G)	post-hoc test results, explanations
Lab assignments	S1. Learning activity	0.618	0.392	0.186	-
	S2. Necessary evil	0.034	0.399	0.547	C: No significant difference on pairwise level. Students in courses C1,C2,C3, and C5 stronger disagreed.
	S3. Stressful	0.057	<0.001	<0.001	P: Students with a lot of prior experience to a larger extent disagreed, sig. differed from students with no (adj. p-value: <0.001) and some prior experience (adj. p-value:<0.001). G: Female students agreed to a larger extent.
	S4. Help from classmates	0.435	<0.001	0.013	P: Students with lots of experience disagreed, and differed from students with some experience, who were more neutral (adj. p-value: 0.06). Students with no prior experience agreed and differed from students with a lot of experience (adj. p-value: <0.001) and from students with some experience (adj. p-value: 0.037). G: Female students agreed to a larger extent.
	S5. Helped classmates	0.097	<0.001	0.228	P: Students with a lot of prior experience sig. differed from students with no (adj. p-value: <0.001) and some prior experience (adj. p-value: <0.001). Students with lots of prior experience to a larger extent agreed.
Exams	S6. Predictable question types	0.031	0.337	0.522	C: No significant difference on pairwise level. Students in C1, C3, and C5 stronger agreed.
	S7. Predictable assessment	0.092	0.428	0.474	-
	S8. Important skills and knowledge	0.008	0.661	0.436	C: Students from course C3 sig. differed from students in C4 (adj. p-value: 0.019), and C6 (adj. p-value: 0.019). C3 students stronger agreed.
	S9. Authentic	<0.001	0.973	0.361	C: C1 differed from C3 (adj. p-value: 0.015) and C5 (adj. p-value: 0.045). C4 differed from C3 (adj. p-value: 0.003) and C5 (adj. p-value: 0.008). C3 and C5 students stronger agreed.
Individual project	S10. Big leap	0.018	0.047	<0.001	C: C1 differed from C5 (adj. p-value: 0.041). C1 students disagreed to a larger extent. P: Not significant on pairwise comparisons. Students with no prior experience stronger agreed. G: Female students agreed to a larger extent.
	S11. Learnt a lot	0.002	0.007	0.350	C: C2 differed from C4 (adj. p-value: 0.002), and C6 students (adj. p-value: 0.015). C2 stronger agreed. P: Students with some prior experience sig. differed from students with a lot experience (adj. p-value: <0.006). Students with some prior experience to a larger extent agreed.
	S12. Authentic	0.003	0.042	0.969	C: C1 differed from C4 (adj. p-value: 0.005). C1 students stronger agreed. P: Students with no prior experience sig. differed from students with a lot of experience (adj. p-value: <0.046). Students with a lot of prior experience to a larger extent agreed.
	S13. Difference between projects	<0.001	0.651	0.718	C: C1 differed from C4 adj. p-value: 0.016) and C5 (adj. p-value: 0.005). (C1 students stronger agreed. C3 also differed from C5 (adj. p-value: 0.028).
Course Coord.	S14. Put their touch	<0.001	0.584	0.971	C: C3 differed from C2 (adj. p-value: 0.014), C4 (adj. p-value: <0.001), and C5 (adj. p-value: 0.036). C1 also differed from C4 (adj. p-value: 0.005). C1 and C3 students stronger agreed.
	S15. Too little insight	<0.001	0.275	0.752	C: C3 differed from C2 (adj. p-value: 0.008), C4 (adj. p-value: 0.004), C5 (adj. p-value: <0.001), and C6 (adj. p-value: 0.001). C1 also differed from C5 (adj. p-value: 0.020). C1 and C3 students stronger disagreed.
Teaching Assistants	S16. Mainly do assessments	0.023	0.266	0.289	C: C2 differed from C3 (adj. p-value: 0.014). C2 students stronger agreed.
	S17. Variation for lab assignments	0.138	0.475	0.283	-
	S18. Variation for projects	0.456	0.475	0.369	-
	S19. Variation for help	0.008	0.158	0.004	C: C3 differed from C1 (adj. p-value: 0.032), and C4 (adj. p-value: 0.004). C3 disagreed while C1 and C4 agreed. G: Female students agreed to a larger extent.
	S20. Variation tutorials	0.055	0.540	0.009	G: Female students agreed to a larger extent.
	S21. Professional	0.224	0.080	0.382	-
	S22. Useful feedback	0.010	0.491	0.027	C: C3 differed from C4 (adj. p-value: 0.005) and C5 (adj. p-value: 0.024). C3 stronger agreed. G: Male students agreed to a larger extent.
	S23. Preferred asking over course coordinator	0.005	0.560	0.887	C: C3 differed from C2 (adj. p-value: 0.019), and C6 (adj. p-value: 0.017). C3 were neutral, while C2 and C6 strongly agreed.
	S24. Preferred internet	0.003	0.425	0.562	C: C3 differed from C1 (adj. p-value: 0.003), and C4 (adj. p-value: 0.014). C3 were more neutral while C1 and C4 strongly agreed.
S25. Too few	<0.001	0.204	0.048	C: C1 differed from C2 (adj. p-value: <0.001), C3 (adj. p-value: <0.001), C4. (adj. p-value: <0.001), and C5 (adj. p-value: 0.009). C1 strongly agreed. G: Female students agreed to a larger extent.	

ZPD and adjusting feedback, support, and instructions accordingly. Previous studies have reported that the students' ZPD can be identified and adjusted within CS courses with promising results [11], [12], [13]. However, more research is needed to provide comprehensive guidelines for how that can be done by TAs within the CS1 context. Strategies for this are beyond the scope of this study, but could be a future research direction.

B. Students' Perceptions of the Assessment Situations

The results showed that the students perceived the lab assignments foremost as a learning activity, having a formative purpose [22]. It implies that the students experienced a value with the assessment that goes beyond the grade, previously described as more desirable [35]. It is a positive finding that the theme "lab assignment as a necessary evil" [7] did not generalize. It also indicates that the students experienced constructive

alignment [14], as the lab assignments were viewed as learning activities but are also part of the summative assessment. Collaborations, pair programming, was encouraged during the lab assignments in most of the courses (all except C1). In these assignments, the social aspect of learning, allowing the students to construct knowledge through discussions [10], has been adopted. The results, however, indicated that not all students experienced being helped or helping each other, possibly due to anxiety as previously found [38]. Some students also experienced stress meeting the deadlines.

Regarding the exam, the students reported that they experienced it as predicable (S6 and S7), aligned with the qualitative findings [7]. Knowing what to expect on the exam could lower test anxiety [37]. It was a positive surprise that most of the students viewed the skills and knowledge assessed by the exam as important, contradicting the findings in the qualitative step. That the exam was experienced as authentic, is surprising since multiple-choice or “fill in the gap” questions are not typically attributed to being authentic. Previously studies have, however, found that students have a rather accurate idea of what authentic means [32]. In the qualitative step, some of the classes used a pen-and-paper exam, and the change of format could also be an explanation for these results since pen-and-paper exams have been experienced as inauthentic [30].

The project was experienced as authentic by most of the students, confirming the qualitative result [7]. These projects are also more similar to real-world problems [34]. The students experienced that they learned a lot from their projects, but also huge differences between the projects regarding how much time was necessary to spend to achieve the same grade, and a big leap in difficulty between the lab assignments and the project. For the students, going from collaborative work with lab assignments to solo programming in a larger project is not trivial. For the students to be successful within their projects, it has to be within their ZPD [9], and they might need help with choosing their projects.

C. Students' Perceptions of the Course Coordinator and TAs

The students experienced that the course coordinator could put their touch on their course, and did not perceive the course coordinator as having too little insights or interest in course activities that were not lectures. These results contradict the findings in the qualitative step [7]. The online format, where it is easier for the course coordinator to move between virtual rooms, could explain this. However, the students perceived their TAs to assess the lab assignments and projects. Not a surprising finding, but even if this task is outsourced to the TAs, the formal assessment responsibility is not the TAs. These results confirm previous claims that the student perceived the TAs to be more approachable than their professors [16]. It also indicates that students were rather comfortable in class with their TAs, which has shown to be a predictor for success [36]. However, a large portion of the students also preferred to search for answers on the Internet rather than asking a TA for help. This might be due to TA availability (S25) or personal preference, but it could also reflect anxiety to ask questions and uncomfortableness [37], [40]. From an instructor's point of

view, you can control what kind of help is provided in class but not what kind of help the students receive on Internet forums.

The students' answers varied regarding differences between TAs, but many had experienced variations. This is alarming since it indicates that not all students experienced getting the same opportunity to learn and receive feedback and that there could be reliability problems with the assessments. Since variation between TAs and uncertainties with the assessments has also been found to be experienced as problematic by TAs [17], [18], it should be addressed as such. Students had also experienced that there could be too few TAs present, and if the assignments are within the students' ZPD [9], they could be too difficult to solve without this help.

D. Differences

While significant differences were found based on students' self-reported previous experience in programming, most of these findings were somewhat expected. The assessments were easier for students with prior experience, and students with no prior experience stated to have gotten more help from their classmates and perceived it as a big leap to move from the lab assignments to the project. These results could make us question if the courses are designed for true beginners, since this group of students seems to have struggled. Although it has been shown in previous research that a subgroup of students tend to struggle and have weak performance in CS1 courses [4]. The grade distribution showed that students with no prior experience did significantly poorer than those with some or lots of prior experience. Naturally, completing a project will be easier for already skilled programmers, however, students with such skills would also be more likely to know which projects are easier and could choose a project strategically. Regardless of prior experience, all students stated that there were differences between projects. Since the learning objectives are the same for all students, it could be argued that all projects should be of similar difficulty for the different grades.

When it comes to gender difference, some results are quite alarming. Female students seem to have a worse experience than their male classmates regarding perceived variations between TAs in tutorials and the amount of help received. Female students also stated there were too few TAs and perceived their feedback as less useful. As shown in a number of previous studies [42], [43], [44], [45], the female students part of this study seems to be in a minority in most of the studied courses, although the gender distribution in the collected responses from C1, C2, and C4 are rather even. This could, however, also be skewed samples since the gender distribution was not collected on the total number of students, only on those that filled out the survey. These differences can not be explained only by differences in prior programming experience (as the other significant differences found, including grade differences). It is reassuring that there was no difference based on gender regarding experienced variation in the summative assessments (S17 and S18). Since CS has been perceived as a male subject [39], the differences identified in the formative assessments could potentially be due to bias and stereotypes. It should, however, be highlighted that

almost all students experienced their TAs to have treated them professionally, important since most of the feedback is given through TA-student dialogues.

These findings align with previous research [46], where avoiding stereotypes, emphasizing collaborative learning, making connections with students' lives and interests, and having meaningful positive interactions with TAs and course coordinators were identified as factors to enable broad participation. It is possible to explain why the female students reported experiencing more stress in labs (S3), receiving more assistance from classmates (S4), and experiencing a wider variation in tutoring quality (S19, S20, S22, and S25) by these learning design recommendations, along with lower confidence, previous background, and a sense of belonging [41], [47], [48]. For further inquiry into this topic, more studies specifically designed to investigate gender differences in assessment situations are needed.

The sample size from the six courses is relatively small, which should be considered when reviewing differences between them. Students in the C3-course (assigned TA, pair programming) was overall more positive to the exam (S8 and S9). As the exam questions were taken from the same pool of questions for all courses, it is likely that this difference reflects overall satisfaction with the course or preparation for the exam. C1 (extra tasks, individual labs graded F–A) students stand out regarding the project, which they seemed to be more prepared for (S10) and to a larger extent experienced as authentic (S12). Possibly since C1 students were encouraged to work alone with the lab assignments. C3 students, for which it was mandatory to pair program, did not differ from C1, indicating the implementation of pair programming has also been successful. In regards to the experience of the course coordinator, C3 and C1 stand out by more strongly agreeing that their course coordinator was able to put their touch on the course and showed interest in all aspects of the course. This indicates that the course coordinators in these courses were more visible to the students. Students in the C3-course had experienced less variations between TAs regarding the formative aspects (S19 and S22) and did not favor searching the Internet for answers over asking the TAs for help (S24). Possibly since C3 students were assigned TAs in set groups, making the students more comfortable. However, this was not seen in C2, in which the students also had assigned TAs. C3 students also experienced their course coordinator to be as approachable as the TAs, while C2 and C6 (labwork in pairs) students experienced the TAs to be more approachable. Students in C1, on the other hand, strongly agreed that there were too few TAs (S25) and preferred to search for information over asking the TAs (S24). In previous studies, TAs have reported having too little time or being understaffed [17], [18] but this was mainly shown in one course (C1).

E. Limitations and Threats to Validity

A limitation, and critique of this study, is that the interview study that informed this data collection was conducted before the outbreak of the COVID-19 pandemic, and the survey data collection began after the outbreak. This means that the learning experience evaluated in the two steps is not identical,

even though the course syllabi and assessment types are the same. Due to the pandemic, most of the education and assessment were conducted online, which could not be controlled due to these unforeseen circumstances.

The low percentage of answers to the survey is a concern and makes the generalizability to the whole student population questionable. It should also be noted that this study has taken place at a single institution and could be context-dependent. In both steps of the data collection, only non-CS majors participated. Hence, it is questionable if these results could be generalizable to other institutions with different organizational setups or to courses that are given to CS majors. However, the mixed-method approach is believed to strengthen the findings within this context. A follow-up study with a larger sample size, or a randomly selected one, could be used to further validate these results. A comparison of the findings to CS majors, could also be a potential future step since that was beyond the scope of this study. For the analysis of the data, future studies should also examine correlations between the prior programming, gender, and course enrollment (or chosen major). In this study, a limitation is that these factors were only compared one at a time.

F. Implications for Practice

Based on the results, lab assignments where students work in pairs, were experienced as a learning opportunity. However, if the students move onto a larger individual, real-world project, the students could need additional support with the transition. Many students reported to having experienced a big leap from the collaborative lab assignments to the larger individual project. Even with the good intention of challenging all students with individual projects, letting the students choose projects of different difficulties is problematic since it can be experienced as unfair as also seen in the results. Instructors who choose this setup are recommended to offer students support in their choice and make sure that tasks of equal difficulty correspond to the same grade. It is also important that the students are supported in choosing a project that is within their ZPD [9], to perceive it as a learning opportunity but not as an impossible task. For individual assignments the course coordinator also needs to take into consideration how the students can receive help, guidance, and feedback from the TAs and instructors. Since it is not a collaborative task, the availability of TAs and instructors is vital for the students to be successful if the projects are within their ZPD.

Although students from all six courses had the same exams, there were course differences. This implies that an exam is not experienced as an isolated event and depends on the courses' learning activities and course structure. The recommendation to course coordinators is to try and integrate the exam with the learning activities and course objectives, in line with constructive alignment [54], and motivate the importance of the skills and knowledge that is tested. Furthermore, if it is important that a CS1 exam is experienced as authentic, and the skills test in such an environment, it should be computer based.

It is positive that respondents perceived their TAs to treat them professionally, but the findings related to variation between TAs are alarming. Furthermore, several gender and

course differences were identified for the formative assessments. To meet the students' need, additional guidance and training for instructors and TAs to support all students in a diverse classroom may be needed. It could also be beneficial to further customize the learning activities, and the formative assessment situations, to the students' individual ZPD [9]. This could, for instance, be done by different layers of scaffolding for the lab assignments based on the student's prior knowledge and skills. Another suggestion would be to actively work with promoting and facilitating a supportive classroom climate [3] and have a variety of themes/topics that the programming assignments are about to promote diversity and inclusion in the classrooms. Since collaborations between TAs have previously shown to have promising results [19], adopting an assessment approach where the TAs work together is recommended to further address differences between TAs. The TA training should also address the use of grading rubrics and offer the TAs guidelines on how they can provide the students with useful feedback, also on an in-course level. The general TA training course should also include material on bias and stereotypes and strategies for promoting supportive and collaborative classroom climates where all students feel welcome and included. In addition, the TAs could be educated to be able to make connect the course material to the students' interest [46]. Of course, this also applies to the course coordinator, who also needs to make sure the course material avoids stereotypes and is recommended to adopt collaborative learning approaches. In light of the finding that many students prefer to search for answers on the Internet, it is also recommended to clarify how sources outside the course are allowed to be used by the students.

REFERENCES

- [1] A. Luxton-Reilly et al., "Introductory programming: A systematic literature review," in *Proc. Companion 23rd Annu. ACM Conf. Innov. Technol. Comput. Sci. Educ.*, 2018, pp. 55–106.
- [2] R. P. Medeiros, G. L. Ramalho, and T. P. Falcão, "A systematic literature review on teaching and learning introductory programming in higher education," *IEEE Trans. Edu.*, vol. 62, no. 2, pp. 77–90, May 2019.
- [3] L. J. Barker and K. Garvin-Doxas, "Making visible the Behaviors that influence learning environment: A qualitative exploration of computer science classrooms," *Comput. Sci. Educ.*, vol. 14, no. 2, pp. 119–145, 2004.
- [4] J. Elarde, "Toward improving introductory programming student course success rates: Experiences with a modified cohort model to student success sessions," *J. Comput. Sci. Coll.*, vol. 32, no. 2, pp. 113–119, 2016.
- [5] L. J. Sax, K. J. Lehman, and C. Zavala, "Examining the enrollment growth: Non-CS majors in CS1 courses," in *Proc. ACM SIGCSE Tech. Symp. Comput. Sci. Educ.*, 2017, pp. 513–518.
- [6] D. Mirza, P. T. Conrad, C. Lloyd, Z. Matni, and A. Gatin, "Undergraduate teaching assistants in computer science: A systematic literature review," in *Proc. ACM Conf. Int. Comput. Educ. Res.*, 2019, pp. 31–40.
- [7] E. Riese and S. Stenbom, "Experiences of assessment in introductory programming from the perspective of NonComputer science majors," in *Proc. IEEE Front. Educ. Conf. (FIE)*, 2020, pp. 1–9.
- [8] K. Falkner and J. Sheard, "Pedagogic approaches," in *The Cambridge Handbook of Computing Education Research*, S. Fincher and A. Robins, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2019, ch. 15, pp. 445–480.
- [9] L. S. Vygotsky, *Mind in Society: The Development of Higher Mental Processes*. Cambridge, MA, USA: Harvard Univ. Press, 1978.
- [10] D. H. Schunk, *Learning Theories: An Educational Perspective*, 6th ed. Boston, MA, USA: Pearson, 2012.
- [11] J. Waite and C. Liebe, "Computer science student-centered instructional continuum," in *Proc. 52nd ACM Tech. Symp. Comput. Sci. Educ.*, 2021, p. 1246.
- [12] A. R. Basawapatna, A. Repenning, and M. Savignano, "The zones of proximal flow tutorial: Designing computational thinking cliffhangers," in *Proc. 50th ACM Tech. Symp. Comput. Sci. Educ.*, 2019, pp. 428–434.
- [13] N. Anderson and T. Gege-Harrison, "Learning computer science in the 'comfort zone of proximal development,'" in *Proc. 44th ACM Tech. Symp. Comput. Sci. Educ.*, 2013, pp. 495–500.
- [14] J. Biggs and C. Tang, "Constructive alignment: An outcomes-based approach to teaching anatomy," in *Teaching Anatomy*, L. Chan and W. Pawlina, Eds. Cham, Switzerland: Springer, 2015, pp. 31–38.
- [15] A. Vihavainen, T. Vikberg, M. Luukkainen, and J. Kurhila, "Massive increase in eager TAs: Experiences from extreme apprenticeship-based CS1," in *Proc. 18th ACM Conf. Innov. Technol. Comput. Sci. Educ.*, 2013, pp. 123–128.
- [16] Y. Ren, S. Krishnamurthi, and K. Fisler, "What help do students seek in TA office hours?" in *Proc. ACM Conf. Int. Comput. Educ. Res.*, 2019, pp. 41–49.
- [17] E. Riese, "Teaching assistants' experiences of lab sessions in introductory computer science courses," in *Proc. IEEE Front. Educ. Conf. (FIE)*, San Jose, CA, USA, 2018, pp. 1–5.
- [18] E. Riese, M. Lorás, M. Ukrop, and T. Effenberger, "Challenges faced by teaching assistants in computer science education across Europe," in *Proc. 26th ACM Conf. Innov. Technol. Comput. Sci. Educ. V.1*, 2021, pp. 547–553.
- [19] B. Harrington, M. Ahmadzadeh, N. Cheng, E. H. Wang, and V. Efimov, "TA marking parties: Worth the price of pizza?" in *Proc. ACM Conf. Int. Comput. Educ. Res.*, 2018, pp. 232–240.
- [20] P. T. Knight, "Summative assessment in higher education: Practices in disarray," *Stud. Higher Educ.*, vol. 27, no. 3, pp. 275–286, 2002.
- [21] L. Allal and G. Pelgrims Ducrey, "Assessment of—or in—the zone of proximal development," *Learn. Instruct.*, vol. 10, no. 2, pp. 137–152, 2000.
- [22] M. Taras, "Assessment - Summative and formative - Some theoretical reflections," *Brit. J. Educ. Stud.*, vol. 53, no. 4, pp. 466–478, 2005.
- [23] A. Jevinger and K. Von Hausswolff, "Large programming task vs questions-and-answers examination in java introductory courses," in *Proc. Int. Conf. Learn. Teach. Comput. Eng. (LaTICE)*, 2016, pp. 154–161.
- [24] A. Cain, "Factors influencing student learning in portfolio assessed introductory programming," in *Proc. IEEE Int. Conf. Teach. Assess. Learn. Eng. (TALE)*, 2014, pp. 55–62.
- [25] D. E. Stevenson and P. J. Wagner, "Developing real-world programming assignments for CS1," in *Proc. 11th Annu. SIGCSE Conf. Innov. Technol. Comput. Sci. Educ. (ITICSE)*, 2006, pp. 158–162.
- [26] J. Sheard, B. Simon, M. Butler, K. Falkner, M. Morgan, and A. Weerasinghe, "Strategies for maintaining academic integrity in first-year computing courses," in *Proc. ACM Conf. Innov. Technol. Comput. Sci. Educ. (ITICSE)*, 2017, pp. 244–249.
- [27] C. Ott, A. Robins, and K. Shephard, "Translating principles of effective feedback for students into the CS1 context," *ACM Trans. Comput. Educ.*, vol. 16, no. 1, pp. 1–27, 2016.
- [28] J. C. Carver, L. Henderson, L. He, J. Hodges, and D. Reese, "Increased retention of early computer science and software engineering students using pair programming," in *Proc. 20th Conf. Softw. Eng. Educ. Train. (CSEET)*, 2007, pp. 115–122.
- [29] J. P. Barros, L. Estevens, R. Dias, R. Pais, and E. Soeiro, "Using lab exams to ensure programming practice in an introductory programming course," in *Proc. 8th Annu. Conf. Innov. Technol. Comput. Sci. Educ. (ITICSE)*, 2003, pp. 16–20.
- [30] P. D. Haghghi and J. Sheard, "Summative computer programming assessment using both paper and computer," in *Proc. 2005 Conf. Towards Sustain. Scalable Educ. Innovat. Informed Learn. Sci. Sharing Good Pract. Res. Exp. Innov.*, 2005, pp. 67–75.
- [31] P. H. Abreu, D. C. Silva, and A. Gomes, "Multiple-choice questions in programming courses: Can we use them and are students motivated by them?" *ACM Trans. Comput. Educ.*, vol. 19, no. 1, pp. 1–16, 2018.
- [32] R. McDermott, M. Zarb, M. Daniels, and V. Isomöttönen, "First year computing students' perceptions of authenticity in assessment," in *Proc. ACM Conf. Innov. Technol. Comput. Sci. Educ. (ITICSE)*, 2017, pp. 10–15.
- [33] R. McDermott et al., "The authenticity of 'authentic' assessment some faculty perceptions," in *Proc. IEEE Front. Educ. Conf. (FIE)*, 2017, pp. 1–9.

- [34] A. Bellino, V. Herskovic, M. Hund, and J. Munoz-Gama, "A real-world approach to motivate students on the first class of a computer science course," *ACM Trans. Comput. Educ.*, vol. 21, no. 3, pp. 1–23, 2021.
- [35] E. Riese, "Students' experience and use of assessment in an Online introductory programming course," in *Proc. Int. Conf. Learn. Teach. Comput. Eng. (LaTICE)*, 2017, pp. 30–34.
- [36] B. C. Wilson, "A study of factors promoting success in computer science including gender differences," *Comput. Sci. Educ.*, vol. 12, nos. 1–2, pp. 141–164, 2002.
- [37] K. Nolan and S. Bergin, "The role of anxiety when learning to program: A systematic review of the literature," in *Proc. 16th Koli Calling Int. Conf. Comput. Educ. Res.*, 2016, pp. 61–70.
- [38] K. Falkner, N. J. Falkner, and R. Vivian, "Collaborative learning and anxiety: A Phenomenographic study of collaborative learning activities," in *Proc. 44th ACM Tech. Symp. Comput. Sci. Educ.*, 2013, pp. 227–232.
- [39] C. M. Lewis, R. E. Anderson, and K. Yasuhara, "i don't code all day": Fitting in computer science when the stereotypes don't fit," in *Proc. ACM Conf. Int. Comput. Educ. Res.*, 2016, pp. 23–32.
- [40] B. C. Wilson and S. Shrock, "Contributing to success in an introductory computer science course: A study of twelve factors," *SIGCSE Bull.*, vol. 33, no. 1, pp. 184–188, 2001.
- [41] K. Redmond, S. Evans, and M. Sahami, "A large-scale quantitative study of women in computer science at stanford university," in *Proc. 44th ACM Tech. Symp. Comput. Sci. Educ.*, 2013, pp. 439–444.
- [42] E. Spertus, "Why are there so few female computer scientists?" Massachusetts Inst. Technol., Cambridge, MA, USA, AI Rep. AITR-1315, 1991.
- [43] J. Margolis and A. Fisher, *Unlocking the Clubhouse: Women in Computing*. Cambridge, MA, USA: MIT Press, 2002.
- [44] V. Barr, "Different denominators, different results: Reanalyzing CS degrees by gender, race, and ethnicity," *ACM Inroads*, vol. 9, no. 3, pp. 40–47, 2018.
- [45] E. Winter, L. Thomas, and L. Blair, "'it's a bit weird, but it's OK'? how female computer science students navigate being a minority," in *Proc. 26th ACM Conf. Innov. Technol. Comput. Sci. Educ. V. 1 (ITiCSE)*, 2021, pp. 436–442.
- [46] B. B. Morrison et al., "Evidence for teaching practices that broaden participation for women in computing," in *Proc. Work. Group Reports Innov. Technol. Comput. Sci. Educ. (ITiCSE-WGR)*, 2022, pp. 57–131.
- [47] B. K. Gjelsten, G. R. Bergersen, D. I. K. Sjøberg, and Q. Cutts, "No gender difference in CS1 grade for students with programming from high school: An exploratory study," in *Proc. 21st Koli Calling Int. Conf. Comput. Educ. Res.*, 2021, p. 5.
- [48] K. Widdicks, A. Ashcroft, E. Winter, and L. Blair, "Women's sense of belonging in computer science education: The need for a collective response," in *Proc. Conf. U.K. Ireland Comput. Educ. Res. (UKICER)*, 2021, pp. 1–7.
- [49] A. Decker, C. Egert, and E. Cascioli, "Cohorting incoming students in a CS1 course: Experiences and reflections from the first year of implementation," *J. Comput. Sci. Coll.*, vol. 35, no. 8, pp. 186–197, 2020.
- [50] J. W. Creswell and J. D. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Thousand Oaks, CA, USA: Sage Publ., 2017.
- [51] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qual. Res. Psychol.*, vol. 3, no. 2, pp. 77–101, 2006.
- [52] E. Riese and V. Kann, "Training teaching assistants by offering an introductory course," in *Proc. 53rd ACM Tech. Symp. Comput. Sci. Educ.*, 2022, pp. 745–751.
- [53] S. L. Weinberg and S. K. Abramowitz, *Data Analysis for the Behavioral Sciences Using SPSS*. Cambridge, U.K.: Cambridge Univ. Press, 2002.
- [54] J. Biggs, "Enhancing teaching through constructive alignment," *Higher Educ.*, vol. 32, no. 3, pp. 347–364, 1996.

Emma Riese received the Ph.D. degree.

She is a Lecturer with the Division of Theoretical Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden. Her research interests are specifically in how summative and formative assessment situations are used and experienced in introductory programming courses in higher education. She is also interested in the interplay between teaching assistants, students, and course instructors.

Stefan Stenbom received the Ph.D. degree.

He is an Associate Professor of Learning in Engineering Sciences with the Division of Digital Learning, Department of Learning in Engineering Sciences, KTH Royal Institute of Technology, Stockholm, Sweden. His research interests include analysis of online and blended learning environments and the community of inquiry theoretical framework.