Getting By With Help From My Friends: Group Study in Introductory Programming Understood as Socially Shared Regulation

James Prather
Abilene Christian University
Abilene, Texas, USA
james.prather@acu.edu

Lauren Margulieux
Georgia State University
Atlanta, Georgia, USA
lmargulieux@gsu.edu

Jacqueline Whalley
Auckland University of Technology
Auckland, New Zealand
jacqueline.whalley@aut.ac.nz

Paul Denny
The University of Auckland
Auckland, New Zealand
paul@cs.auckland.ac.nz

Brent N. Reeves
Abilene Christian University
Abilene, Texas, USA
brent.reeves@acu.edu

Brett A. Becker
University College Dublin
Dublin, Ireland
brett.becker@ucd.ie

Paramvir Singh
The University of Auckland
Auckland, New Zealand
p.singh@auckland.ac.nz

Garrett Powell
Abilene Christian University
Abilene, Texas, USA
gbp18a@acu.edu

Nigel Bosch
University of Illinois
Urbana–Champaign, Illinois, USA
pnb@illinois.edu

ABSTRACT

Background and Context. Metacognitive skills are important for all students learning to program and interest in applying pedagogical approaches in early programming courses that focus on metacognitive aspects is growing. However, most studies of such approaches are not rigorously based in theory, and when they are, almost always utilize foundational education and psychology theories from as far back as the 1970s. More recent theory is less tested, and not all relevant metacognitive theories have been explored in the computing education research literature.

Objectives. We present the first use in a programming education context of a newer metacognitive theory that explicitly examines the differences between self-regulation, co-regulation, and socially shared regulation. Our research questions are: 1) How do students express their learning strategies, both when working alone and when working in groups, and how do these align with existing models of self-regulation and co-regulation? and 2) To what extent do written expressions of self-regulation, co-regulation, and socially shared regulation relate to student performance?

Methods. Grounded in the above mentioned theory, we collected qualitative self-reflection and quantitative course performance data from nearly 1,000 students in an introductory programming course. We use these data to explore students' self-regulation habits when studying alone and their co-regulation habits when studying in groups.

Findings. Our findings indicate that higher self-regulation correlates with higher performance, but higher co-regulation had the opposite effect. We explore these differences through a qualitative analysis of the self-reflection statements and identify co-regulation strategies to build upon existing models of self-regulation.

Implications. We identify emergent themes in our data that align with those in recent literature in self-regulated learning in computing education and present the first set of co-regulation themes in computing education. This work is at the frontier of self- and co-regulation in introductory programming and identifies several factors that can be used to advance future work and, most importantly, improve student outcomes.

CCS CONCEPTS

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

KEYWORDS

groups; group work; CS1; introductory programming; programming education; metacognition; self-regulation; co-regulation; socially shared regulation; studying; study habits

ACM Reference Format:
James Prather, Lauren Margulieux, Jacqueline Whalley, Paul Denny, Brent N. Reeves, Brett A. Becker, Paramvir Singh, Garrett Powell, and Nigel Bosch. 2022. Getting By With Help From My Friends: Group Study in Introductory Programming Understood as Socially Shared Regulation. In *Proceedings of the 2022 ACM Conference on International Computing Education Research V.1 (ICER 2022), August 7–11, 2022, Lugano and Virtual Event, Switzerland*. ACM, New York, NY, USA, **??** pages. https://doi.org/10.1145/3501385.3543970

ICER 2022, August 7–11, 2022, Lugano and Virtual Event, Switzerland

ACM ISBN 978-1-4503-9194-8/22/08.
https://doi.org/10.1145/3501385.3543970

1 INTRODUCTION

Interest in metacognition from computing education researchers has increased over the past few years [43]. Metacognition – thinking about thinking – is a higher-order cognitive skill that has been explicitly linked to the ability to think computationally [57]. However, for tasks that demand high cognitive load to process domain knowledge, like learning to code, simultaneously processing metacognitive knowledge and strategies can be unattainable, especially without explicit guidance [37]. As a result, some have advocated for explicit teaching of metacognition in early computing courses [31, 32]. Others have attempted to introduce metacognitive scaffolding as an implicit element in the course [13, 14, 44–46]. There are also facets to programming courses that take place outside the classroom, such as students studying in groups. However, most research on metacognition centers the learner as an individual without accounting for the interactions that students have with their peers while learning, particularly outside direct instruction.

Collaboration and group work have been studied extensively in the computing education literature [21, 23, 29, 48], but few have sought to place it in the context of metacognition [58]. To address this gap, we apply Hadwin et al.'s Socially Shared Regulated Learning (SSRL) model to programming education [24]. This model delineates between the different types of self-regulation in social learning environments, including: 1) how learners manage their own self-regulation; 2) how learners help others self-regulate; and 3) how learners in groups regulate each others' metacognition.

In this paper, we present an empirical study on student study habits and explicitly link these habits to a modern theory of social metacognition. We collected data from a large introductory programming course ($n \sim 1000$) by asking students at two different points in the course to reflect on their own study habits and how they study in groups. As the first application of the SSRL model in programming education, our work focused on an exploratory thematic analysis of these qualitative data as well as quantifying them through an automated analysis that detected the use of metacognitive phrases (e.g., "I didn't realize...", "I know about..."). We also collected student performance data and Likert-type questions about self- and social-regulation. Our quantitative analysis correlates performance with the metacognitive statements extracted from their written reflections. The qualitative analysis explores the growth and change in the self-regulation and socially shared regulation of the students from their initial statements to their final reflections. We seek to answer the following research questions:

- RQ1 How do students express their learning strategies, both when working alone and when working in groups, and how do these align with existing models of self-regulation and co-regulation?
- RQ2 To what extent do written expressions of self-regulation, co-regulation, and socially shared regulation relate to student performance?

This work has three important contributions. First, we validate previous work on self-regulation in programming by situating our findings in multiple existing frameworks from the literature. Second, we provide the first exploration of naturally occurring study group behavior through the lens of Hadwin et al.'s SSRL model. Third, we provide themes for co-regulation in programming similar to those already provided in the literature for self-regulation in programming.

2 RELATED WORK

In this section, we examine the related literature around metacognition and self-regulation, its incorporation into introductory computing courses, and study behavior in groups in these same courses.

2.1 Metacognition and Self-Regulation

There are two primary foundational theories of metacognition from the field of psychology. The first comes from Flavell in 1979 [19], which he described as cognition that regulates any aspect of cognitive control. Although Flavell's theory is not specifically about (or in the context of) education, researchers have extensively applied his theory to the domain of education [6]. A second theory comes from Bandura in 1986 and focuses more on the cognitive process itself, which he termed self-regulation [2]. Bandura proposed that when engaged in a task, a person also cyclically engages in self-observation and evaluation of progress towards goals, adjusting behavior based on this evaluation. This cycle of metacognitive thought and action proved to be foundational as several subsequent theories built upon it [7, 15, 40, 60].

The theories described above appear commonly in computing education research in addition to two others that are based on them. The first is Zimmerman's theory of self-regulated learning, which builds on Bandura's theory by adding metacognitive knowledge that appears as a forethought phase. The second is Pintrich's theory that splits phases of regulation from areas of regulation in each phase [41]. Pintrich also adds additional self-regulated learning phases to allow a more fine-grained approach to modeling the interactions between observable phenomena, such as between monitoring and control. [40].

In the past decade, the theories of Flavell, Bandura, Zimmerman, and Pintrich, which are also popular outside of computing education, have been appearing with greater frequency in computing education research literature [43]. However, newer and more specific theories have been put forward that may be more appropriate to use compared to their earlier and more general predecessors [33]. For instance, a self-regulated learning theory proposed in 1998 by Winne and Hadwin [55] appears to be particularly well-suited for the task of programming. This saw its first use in computing education research literature in 2022 [11]. The SSRL theory by Hadwin et al. [24] mentioned above, which incorporates regulation of cognition in others and between others, represents the cutting edge of this family of theories and was most recently updated in 2018 [33]. The SSRL model specifically adds co-regulation and shared-regulation to Winne and Hadwin's model. In social settings such as study groups, people are helping to support the regulation of others' cognition. At the same time, study groups are working together to plan what they will study, select strategies to complete their goals, and evaluate the success or failure of those goals. This is the first model to explicitly apply the process of self-regulation to interaction with others and groups. In this paper, we provide the first applications of Hadwin's SSRL model to naturally occurring group study behavior in an introductory programming course.

2.2 Metacognition and Self-Regulation in Introductory Programming

Two comprehensive reviews of metacognition and self-regulation theories in computing education research have recently been published [33, 43], which we summarize here for this paper's context. Work before 2010 is quite scarce, though there are some notable exemplars [4, 39]. In 2011, VanDeGrift et al. argued that programming teachers should not only teach programming but also metacognitive skills that will help students assimilate new knowledge going forward [52]. Since then, a wealth of studies have tackled the subject [17, 27, 28, 31, 32, 38, 44, 45, 53]. In particular, there are two studies highly relevant to our research questions that we detail below.

Falkner et al. used Zimmerman's theory to identify self-regulated learning strategies specific to the domain of learning computing [17]. They asked 85 students to reflect on their software development process and coded all responses, grouping them by common themes. By far the most common group of themes was around the development process itself, followed by decomposing the problem, time management, assessing difficulty, and building knowledge. Some of the specific strategies in those themes are very relevant to our research questions involving how students regulate themselves and their peers in study groups. These include "use diagrams to describe or explain design", "use design to understand problem or code", "create plan from design", "prioritisation", "design as aid to time management", and "practice writing code". We look for Falkner's themes appearing in our dataset in the discussion below.

Loksa et al. framed the process of solving a programming problem as an iterative series of problem-solving stages that, when explicitly taught to students, can increase their metacognitive awareness [32]. These problem-solving stages are: 1) reinterpret the problem prompt, 2) search for analogous problems, 3) search for solutions, 4) evaluate a potential solution, 5) implement a solution, and 6) evaluate the implemented solution. Students in an experimental group were given a handout with these problem-solving stages and when asking for help were prompted to verbalize in which stage they thought they were stuck. Loksa et al. found that by explicitly scaffolding the novice coder experience, students in the experimental group were more able than students in a control group to self-regulate their own problem-solving process. Their problem-solving framework has been utilized in computing education contexts to further explore metacognitive issues novice programmers face when learning to code [13, 44, 45]. We therefore expect to find these problem-solving stages in student reflection responses pertaining to both their own self-regulation as well as that of their group.

2.3 Collaboration and Group Work

Group work and collaborative learning in introductory programming (often but not always tied to CS1) has been an important topic in computing education research since the 1990s [3, 12, 16, 18, 21, 23, 29, 35, 48, 54]. However, even very recent work on the topic usually does not discuss it in terms of its impact on metacognition and self-regulated learning [56]. In a comprehensive literature review of self-regulation interventions in programming education, Silva et al. found no studies that have looked at social interactions in novice programming education through the lens of socially shared regulation of learning [48]. However, Silva is currently working on a dissertation that utilizes Hadwin's SSRL model to better understand how social interactions in programming education tasks impact self- and shared-regulation [47, 49].

A small number of researchers have looked more closely at the relationship between metacognition and computer-supported collaboration in programming. Bachu and Bernard discussed the impact of collaboration on novice programmer metacognition and how it benefits the problem-solving stages of programming [1]. They used games and social networking to provide the computer-supported collaborative learning aspect [5]. Their intervention was correlated with increased performance on programming tasks and saw an increase in metacognitive behaviors associated with programming problem-solving. Unlike Bachu and Bernard, the present study attempts to determine if naturally occurring group study behaviors impact self and shared regulation.

Beyond the field of computing, researchers are exploring how the theory of Hadwin et al. might apply to their discipline-specific educational settings [20]. For instance, Hurme et al. discuss what makes computer-supported collaborative problem-solving in mathematics a socially shared phenomenon [26]. Zheng et al. recently conducted an experiment with physics students to determine if group metacognitive scaffolding exercises impacted metacognitive behavior and group performance [58]. They divided students into groups of three, assigning half the scaffolding exercises, before completing a collaborative online learning assignment. Their results indicate that group metacognitive scaffolding can significantly increase metacognitive behavior and performance when collaborating online. The present study connects this work in collaboration and group work, through the lens of Hadwin's SSRL theory [24], to introductory programming classes, especially those with online components.

3 RESEARCH METHODS

We employ a mix of quantitative and qualitative research methods in this study. Our data are from a first-year undergraduate engineering course. The details about the students, their course and its context are presented in section 3.2.

3.1 Summary of Methods

In addition to collecting student marks from the course as a proxy for performance, we used a repeated-measures design, asking all participants to complete the same questionnaire at the midpoint and end of the course. The study did not employ an intervention for self-regulation or socially shared regulation. These data collection points were intended to evaluate whether strategies changed throughout the semester. The questionnaire had eight questions (six quantitative and two qualitative). Half of the questions were about self-regulation (three quantitative, one qualitative), and the other half were about socially shared regulation.

The quantitative questions were taken from the Metacognitive Self-Regulation and Peer Learning subscales of the Motivated Strategies for Learning Questionnaire (MSLQ) [42], one of the most used self-regulation instruments in education [43]. The original MSLQ uses a Likert-type scale (7-point with the anchors 'not at all true of me' to 'very true of me'). Questions about co-regulation also included an additional 'Not applicable, I do not study in groups' option. The full subscales (12 questions each) were too long for this context, so we selected the most relevant three items from each (e.g., we did not include questions about reading strategies). The three quantitative questions for each type of regulation (S1–S3 in

Figure 1 and C1–C3 in Figure 2) were also intended to prime students to expand upon their regulation strategies in the open-ended qualitative questions (S4 and C4 in Figures 1 and 2), which form the bulk of our analysis.

Like any subject, learning programming presents unique challenges. Everyone learns and studies in their own way and now, as you begin learning a new programming language, it may be beneficial to reflect on what strategies work best for you. The next few exercises will step you through this reflection.
"Self-regulation" of learning describes the processes that help you understand what is working or not about your behavior and strategies for learning. Please answer the following questions about your self-regulation of learning. Being as honest as possible is important for you to benefit the most from this reflection task.

- S1 During class, watching recordings, or self-studying, I often miss important points because I'm thinking of other things.
- S2 I ask myself questions to make sure I understand the material I have been studying in this course.
- S3 When I study for this class, I set goals for myself in order to direct my activities in each study period.
- S4 With these questions in mind, write a short reflective piece (anywhere from several sentences to several paragraphs) in your own words, that describes how you study for this course, regulate your own learning, and what you find works best for you. What study techniques do you find are most effective and work best for you when learning programming in this course?

Figure 1: Reflective prompt and response options targeting self-regulation

In the course, any source code that you submit for marking should be written by yourself, however discussing ideas at a "high-level" or talking through problems with others can be helpful. You are encouraged to discuss course material or general problems with others, if you find that useful. Like "self-regulation" of learning when you study by yourself, when you study with others or in a group the term **"co-regulation"** of learning describes the social strategies and processes that you use when learning together. Please answer the following questions about your preference for working with others and 'co-regulating' your learning with other students.

- C1 I study for this course with a friend or a group of peers, rather than study by myself.
- C2 When my study partners and I become confused about something we are studying for in this course, we go back and try to figure it out together.
- C3 When my study partners and I study for this course, we set *group* goals for ourselves in order to direct our activities in each study period.
- C4 With these questions in mind, write a short reflective piece (anywhere from several sentences to several paragraphs) in your own words, that describes how you study for this course in groups or with others, co-regulate the group's learning, and what you find works best for your group. What study techniques do you find are most effective and work best when learning programming together? If you don't ever study in a group you can leave this question blank.

Figure 2: Reflective prompt and response options targeting co-regulation

3.2 Course Context & Data Collection

Our data were collected from a cohort of first-year engineering students at the University of Auckland, a large public research university in New Zealand. All students in the engineering program at this institution take a compulsory programming course which covers two programming languages, MATLAB and C, taught in consecutive 6-week modules. There is a two-week break between the two halves of the course. The study was conducted during the 2021 edition of this semester-long course, with a total of 1,081 students enrolled. The course comprised of five weekly contact hours including three hours of lectures and two hours of laboratory tasks.

Students were invited to complete all of the reflective questions shown in Figures 1 and 2 on two occasions. The first was in Lab 7 (the first lab of the second module) and the second was in Lab 12 (the final lab of the second module). This design was to elicit reflections that looked back over both modules of the course. All C module lab (programming) exercises were completed using CodeRunner [30], which was also the platform used to collect data for this study.

3.2.1 Validity of Data during the COVID-19 Pandemic. Lectures and labs are usually conducted in-person on campus. In 2021, however, due to restrictions related to the COVID-19 pandemic, the course was moved to online delivery mode at the end of Week 5. This meant a total of 15 lectures and 5 labs, i.e., Labs 1–5 (inclusive), were conducted on campus, and the remaining 21 lectures and 7 labs, i.e., Labs 6–12, were conducted online. Thus, the whole of the C language module of the course was delivered online. In this online mode, the delivery of the course material was altered with pre-recordings of lecture content shared at the start of each week. Students could optionally attend live help sessions via Zoom to receive support from teaching assistants, similar to the support usually offered on campus. Additionally, the Q&A platform Piazza was available, and used extensively by students during the course.

To evaluate how these instructional media changes affected the validity of our data, we compared student performance in 2021 to that in 2019, before the pandemic, and in 2020, when the first shift to online instruction occurred due to the pandemic. Based on these comparisons, we found 2019 and 2021 to be much more similar than either was to 2020. Pillai's trace, the most commonly used inferential statistic for multivariate ANOVA models, ranges from 0 to 1, with larger numbers interpreted as stronger evidence of a difference between groups. Comparing 2019 to 2021 gives a Pillai's trace of 0.04, while comparing 2020 to 2019 gives 0.14 and comparing 2020 to 2021 gives 0.09. Given a sample size of over 1,000 for each year, all differences were statistically significant, but comparisons to 2020 are two- to three-times larger. Further, when comparing performance between the MATLAB and C portions of the course, there is little variance within students in 2019 and 2021, $F = 0.06$, $p = 0.81$. However, differences within students were substantial when comparing 2020 to 2019, $F = 15.54$, $p = <0.01$, and 2020 to 2021, $F = 17.30$, $p = <0.01$.

Looking at data distributions, the first pandemic-based shift in 2020 to online mode had various effects on students. Examining performance before and after the date of the shift, some performed worse, some the same, and some better. This type of unpredictability did not appear in 2021 data, but one large difference between 2019 and 2021 can account for most of the differences found. In 2021, students had an extra week to work on the MATLAB project due to the timing of the shift to online. Likely as a result, marks on the MATLAB project were 15% higher in 2021 than in 2019. Otherwise, performance in 2019 and 2021 is indistinguishable on an aggregate scale. As a result, we argue that the data analyzed in the current study are not significantly affected by pandemic-related effects.

3.3 Performance Data

Assessed components of the course included assignments from the MATLAB and C modules, which each had weekly labs, one programming project, and one test. In addition, there was a final exam at the end of the course. The first metacognition survey was given during the first week of the C module and, thus, primarily describes students' strategies during the MATLAB module. In this course, most of the MATLAB module was face-to-face. The second metacognition survey was given near the end of the course and, thus, primarily describes students' strategies during the C module, which was online. Accordingly, we correlated data from the first survey with MATLAB performance plus the final exam, and we correlated data from the second survey with C performance plus the final exam. Weekly lab performance was ultimately excluded from analysis because these data had a high mean, high kurtosis, and high skewness, suggesting that there was a ceiling effect that would provide little distinction between students.

3.4 Thematic Analysis of Open Response Questions

A subset of the student responses to questions S4 and C4 (see Figures 1 and 2) at the two time points, Week 7 and Week 12, were analyzed using a thematic analysis approach [9]. Responses from the same randomly selected students were used for all four instances of the open questions. Three of the authors performed an open coding of the student responses. First, the Week 7 responses to S4 were coded, with new codes being added as they were encountered until the three coders agreed that they had reached saturation. We applied a similar interpretation to that of Given [22], with saturation being the point at which 'additional data do not lead to any new emergent themes'. The resulting set of codes was used as a starting point when coding the later Week 12 responses to S4 in order to enable the comparison of codes at these two time points. The same process was repeated for the Week 7 and Week 12 responses to C4.

The emergent codes were then merged by a process of discussion and consensus to develop a set of themes related to self-regulation (S4) and co-regulation (C4). These themes were then mapped to those reported in the literature.

3.5 Extracting Metacognitive Language Use from Open Response Questions

We also analyzed the open response questions via a natural language processing method designed to extract expressions of metacognition in student-written text [25]. This method recognizes statements of knowing, not knowing, and related statements of certainty regarding knowledge. In particular, the method recognizes statements made in first person singular or plural, which thus describe one's own cognition (i.e., a metacognitive statement) either alone (e.g., "I...", "my...") or in a group (e.g., "we...", "our..."). Such statements are primarily denoted by a word indicating metacognition (e.g., "understand", "considered"), though the method also considers some statements that do not follow the most common *pronoun...metacognitive indicator* form, such as "it made me realize". The method has been validated with secondary and postsecondary school students [8, 25], including in computer programming contexts [11]. Thus, we expect it is appropriate for the data in this study. We utilized this method as a way of validating the frequency of themes identified by our qualitative analysis.

4 RESULTS

4.1 Quantitative Analysis of Performance and Surveys

To explore the relationship between performance and students' responses on Likert-type questions for self- and co-regulation, we utilized correlation as an inferential statistic. We did not attempt to affect self- and co-regulation via an intervention, so correlation is the most appropriate statistical analysis to explore their relationship to performance. We use Pearson's correlation coefficient because both the survey and performance values are continuous, rather than discrete, variables. Out of 1,081 students, 963 provided complete responses. Examining the distribution of data, we found no evidence of bias in which students had complete data, either in terms of performance in the course or responses to Likert-type questions. That is, students who failed the course or reported low self-regulation were no more likely to have incomplete data than those that achieved the highest performance or reported high self-regulation.

There was a consistent, small, positive relationship between self-regulation (the sum of responses to S1–S3, with S1 reverse-scored) and performance (see Table 1), as we would expect based on the literature [4]. In other words, students who reported engaging more in self-regulation had higher marks. For co-regulation (the sum of responses to C1–C3) we found no consistent relationship with performance (see Table 1) or self-regulation (r = 0.07, p = 0.08). The only significant correlations between co-regulation and performance was with the C project and the final exam, which had negative correlations. These results suggest that students who said they engaged in more co-regulation tended to do worse on the C project and final exam. These relationships are explored more in the qualitative analysis.

The most likely explanation for the negative relationship between co-regulation and performance in the latter half of the class is the difficulty of working in groups during the online portion of the course. Although we did not observe a large drop-off in group

Table 1: Correlations between Performance and Self- and Co-Regulation Likert Responses for MATLAB and C Modules.

Module	Likert	Test		Project		Final Exam	
		r	p	r	p	r	p
MAT-LAB	Self-reg (S1–S3)	0.09	<0.01	0.09	<0.01	0.11	<0.01
	Co-reg (C1–C3)	-0.06	0.08	-0.05	0.16	-0.06	0.10
C	Self-reg (S1–S3)	0.07	<0.05	0.13	<0.01	0.14	<0.01
	Co-reg (C1–C3)	-0.05	0.20	-0.09	0.03	-0.13	<0.01

work before and after the shift to online learning, 242 students on the first survey and 283 students on the second survey reported that they 'don't study in groups', which was an option in addition to the Likert-type scale for C1–C3. The reasons that students describe for working in groups from the qualitative analysis is informative. We found that students who performed well did not rely on peer help while students who struggled would continue to reach out to their peers for help during the online portion of the course. Thus, co-regulation behaviors during the online portion of the course became somewhat of a proxy for students having difficulty with the coursework.

4.2 Qualitative Thematic Analysis

We coded responses from 92 students (approximately 9% of those enrolled). Across prompts S4 and C4 at both time points, responses from these students totaled 18,576 words. Table 2 provides a descriptive summary of the word length of responses to each prompt and the total number of distinct codes that emerged during thematic analysis and were used for coding the responses to each prompt.

Table 2: Word length (mean and SD) for responses to S4 and C4 at Week 7 and Week 12.

Time	Prompt	Mean	SD	Codes
Week 7	Self-regulation (S4)	72.4	70.1	46
	Co-regulation (C4)	47.1	50.0	44
Week 12	Self-regulation (S4)	50.4	35.5	58
	Co-regulation (C4)	36.3	36.1	43

In general, students provided longer responses to the Week 7 prompts than to the Week 12 prompts, and also tended to provide longer responses for the self-regulation prompt compared with the co-regulation prompt. Both of these observations match our expectations. Students tend to be busier with coursework around the final week of the course, and thus have less time for activities they perceive as not providing direct benefit in terms of learning programming. In addition, it is natural to expect that all students have developed strategies for learning on their own, but not all students prefer to work in groups.

The number of codes that were used to characterise the co-regulation responses was consistent between Weeks 7 and 12. In contrast, a greater variety of codes were used to describe the responses to the self-regulation prompt in Week 12 compared to Week 7 (see totals in Table 3). A total of 58 distinct codes were used for the Week 12 responses, despite these responses in general being shorter, compared to 46 codes used for the Week 7 responses. Potential explanations for this change are a modification of learning strategies, or reflection on the value of different strategies, caused by a shift to online teaching as a result of the pandemic. This shift occurred in the penultimate week of the first course module, meaning that when students responded to the first reflective prompt in Week 7, most of their learning had been in-person and on campus. When students responded to the second reflective prompt in Week 12, their learning experience had been entirely online since their initial reflection. Some codes explicitly related to the shift online (i.e. lockdown), such as: 'lockdown affected my schedule/progress' and 'watch videos online due to lockdown' and these were classified under the 'online learning - lockdown' theme which emerged only in the Week 12 reflection. We observed some comments, also under the same theme, that indicated students' strategies were greatly impacted by the lockdown. The following student quote epitomizes this sentiment:

> I found having a proper schedule really effective. I had a very good schedule - watched all lectures, handed in labs early before lockdown occurred. Once lockdown occurred my schedule sort of collapsed and hasn't been able to get back on track.

Other codes were less explicit about the lockdown, but referred to resources that were only available during the second course module which included 'drop-in online help sessions' that were provided as a result of the shift online.

A total of 76 distinct codes were used across the responses to the self-regulation prompt in both Week 7 and Week 12, and a total of 50 codes were used across responses to the co-regulation prompt. These codes were refined to a set of broader themes that captured the various strategies students reported using when learning in the course. Table 3 lists the most commonly occurring themes. Some responses were quite long, and assigned more than one code that ended up mapping to the same theme. An example of this is for the theme 'coding practice', where the initial set of emergent top-level codes included 'working on lab exercises', 'working on practice problems' and 'playing/experimenting with code'. Some students gave responses that were tagged with more than one of these codes. In Table 3, we report the themes for both the self-regulation and co-regulation prompts, and the number of students who provided a response that was coded with respect to each theme. The table is sorted by the frequency of each theme for the Lab 7 responses.

Since we randomly sampled 92 students out of 962, we utilized the NLP tool described in Section 3.5 to determine if this sample of students was providing responses that were consistent, at least with respect to expressions of metacognition, with those provided by the entire cohort. To test this, we conducted a series of two-sample t-tests assuming unequal variance between the participants we randomly sampled for our qualitative analysis and those not sampled. The dependent variable in this case was the number of recognized statements that describe one's own cognition. Table 4 provides the results of these tests. In each case, since $p > 0.05$, we fail to reject the null hypothesis that the difference in group means is zero. We interpret this finding as support that the 92 participants randomly sampled are a representative subset and our qualitative findings can be generalized across the student cohort.

Table 3: Most commonly occurring themes in the responses to the self-regulation and co-regulation prompts for Labs 7 and 12. Frequency counts are the number of students who gave a response that was coded with the corresponding theme.

	Frequency			Frequency	
Self-regulation theme	**Lab 7**	**Lab 12**	**Co-regulation theme**	**Lab 7**	**Lab 12**
Coding practice	54	65	Help seeking - social	35	35
Resources helpful	45	52	Group learning	22	31
Self-explanation	17	10	Learn through teaching	7	10
Goal-setting and planning	10	15	Resources helpful	7	3
Help seeking - information	9	7	socially shared regulation	6	1
Time management	6	11	Group averse	5	6
Help seeking - social	5	6	Group planning	3	6
Environmental restructuring	4	1	Help seeking - time management	3	4
Self-evaluation	2	10	Help seeking - information	3	2
Motivation	2	4	Motivation	2	2
Resources unhelpful	1	4	Self-explanation	2	0
Rule of thumb	1	0	Online learning - lockdown	1	6
Online learning - lockdown	0	5	Coding practice	1	2
Visualization	0	2	Environment	1	0
Total	156	192	Total	98	108

Table 4: Results of statistical tests comparing metacognitive count from the NLP tool between participants randomly sampled (n=92) and those not sampled (n=870).

Question	p	t	df
Lab 7 S4	0.077	1.79	80
Lab 7 C4	0.071	1.82	78
Lab 12 S4	0.817	0.23	87
Lab 12 C4	0.309	1.02	86

4.2.1 Self-regulation Themes. Responses to the prompt about self-regulation were dominated by statements that described the benefits of practicing writing code and the concrete resources that students found helpful. The course includes weekly lab sessions, and these were frequently mentioned as being valuable for providing opportunities to practice writing code (theme 'coding practice'):

- "My main method for studying for this course is to just practice a lot for the coding."
- "I think the way I will study best is by practicing writing code as much as possible, so basically doing as many practice questions as I can."
- "The way I study for this course is by practising, doing labs. Practising programming works the best for me, as I'm thinking of the problem, how to solve it, then implementing it."

Interaction with standard course resources, such as watching lectures (with more explicit mention of recorded or online lectures in the Week 12 responses), reading the coursebook, and studying past tests and exams all appeared frequently in student responses. In the majority of cases, these were viewed favourably by students, with 150 codes relating to the course resources being helpful for learning (theme 'resources helpful'):

- "I will mainly rely on reading and understanding the course book to learn how to complete lab tasks/assignments."
- "I find the best way of learning is to listen to the lectures without taking notes. That way I can fully listen to the new content being taught and make sure I have a solid grasp of the concepts."
- "In regards to exam study, I find it beneficial to look through the past tests in my scheduled time."

In a small number of cases, students noted that some resources were not helpful to them. These included statements that lectures were confusing, or that watching recorded lectures is time consuming. This was one of the least common themes emerging from the data. However, given that prompt S4 explicitly asked students to comment on the strategies that they find work best for them, this is perhaps unsurprising (theme 'resources unhelpful'):

- "I found many of the lectures too confusing to follow so most of my learning was conducted myself when going through the labs."

Various strategies were reported for setting goals, planning and managing time. These included a range of strategies from quite systematic approaches for spreading work over time to more general comments around setting daily goals (themes 'goal-setting and planning' and 'time management'):

- "Something that works for me is to know what I'm going to do when I wake up tomorrow, in my head I will think about what I need to do and what I want to get done before a certain time.e.g if I have 4 lectures to watch that day, ill try watch the first one before 10, the second before 12 etc."
- "I regulate my learning by making a list of goals to complete each day."
- "I regulate my learning by planning the amount of work/tasks I want to complete within the specified time. I record how

much time it takes me and reflect to prepare and adjust for the next task."

Almost all strategies relating to goal-setting and planning or time management appeared to be productive ones. However, one student acknowledged that being particularly stubborn is not a healthy strategy as it can negatively impact their time management (theme 'time management'):

- "Although it might not be the healthiest strategy, when I have coursework to do I typically sit down at my desk and don't leave until I've completed the task, like I did with this lab. Sometimes I don't even notice the time going past, as I get absorbed in whatever the task is, but other times I spend hours stuck on one thing and am too stubborn to move on."

4.2.2 Co-regulation Themes. In response to the co-regulation prompts, 57 students gave responses that either explicitly or implicitly stated that they tended to work alone (we report the frequency of such responses here, however we did not classify these under a co-regulation strategy). Some of these responses also expanded on strategies for when they did work with others, in which case they were coded. This tendency to work alone may have been impacted by the difficulties that the shift to online learning presented in terms of not being co-located with other students on campus. Indeed, only eight such statements were observed in the Week 7 responses, with the remaining 49 being from the Week 12 responses.

The most common theme emerging from the co-regulation prompts was around social help seeking, and this appeared much more frequently than in response to the self-regulation prompts (theme **'help seeking - social'**):

- "When we try hard but still can not figure out how to solve the problem, we tend to ask each other. There are always someone knows how to solve it."
- "When I am stuck, and don't know what to do even after reviewing what we've learnt, I would ask my friends, who are all quite capable. Instead of just giving me their work (which is completely and utterly unacceptable plagiarism), they guide me through the question, and tell me what I did wrong."

Group learning emerged as another popular theme. Individual codes corresponding to this theme were assigned to statements relating to groups discussing approaches and concepts together, and sharing learning strategies with each other (theme **'group learning'**):

- "I find it effective to work in a group when studying as we can bounce ideas off each other and help each other understand concepts."
- "I like to interact with a group of people where we can all share how we solved the prep task or how we understood something so we understand better."

Several new themes that related to working with others emerged from the co-regulation responses. For example, we observed a small number of responses that described the groups taking metacognitive control of certain tasks, monitoring progress and regulating the behavior of group members (theme **'socially shared regulation'**):

- "The group members will supervise each member's learning progress to ensure that everyone is not left behind and that no individual will slow down the group's learning progress."
- "The majority of the 'group' study is more us keeping each other on track, reminding each other to study sometimes being on call, not helping each other or discussing what we are studying but just then we can keep each other accountable and make sure we both study productively."

Another new theme that emerged in the co-regulation responses was that of learning through teaching (theme **'learning through teaching'**):

- "I find that one of the best ways to cement the things I have learned is to teach it to others."
- "This meant I was able to help my other friends helping to reinforce my knowledge while letting me notice any cracks I have in my knowledge. In this way its a win - win solution."

For those students who tended to work alone, when reasons were given they were commonly to avoid distractions or to have the ability to work at their own pace (theme **'group averse'**):

- "I've never had the luxury to study in groups or a group. I always end up doing things on my own at my own pace where I am most comfortable."
- "I'm a person that can get distracted with others, so I prefer to study individually by myself."

As expected, there were many blank fields in the co-regulation open response. However, some students provided interesting input, such as one who simply wrote "I don't really study in groups. Don't have friends lol" for Lab 7. In Lab 12, the same student wrote "Still don't have friends." Another student wrote that they prefer to ask the internet for help and added: "I feel embarrassed asking my friends why my code is not working."

5 DISCUSSION

In this section, we first present the overlap of our findings on self-regulation with those in the literature mentioned in Section 2.2. Our first contribution is to validate the findings of previous work by showing similar emergent themes with regard to self-regulation in programming. We then discuss our novel findings on co-regulation in programming and demonstrate that our findings correspond to the SSRL model by Hadwin et al. [24].

5.1 Self-regulation Themes

The emergent themes based on our qualitative analysis of the open questions exploring student self-regulation were found to have some alignment with those reported in the literature as shown in Table 5. Four of our six most-frequent self-regulation themes overlap with the CS-specific themes found by Falkner et al. [17]. Five of the six programming self-regulation behaviors recorded by Loksa et al. [34] aligned with our themes. In addition, many of the more general self-regulation behaviors originally detailed by Zimmerman also appeared in our dataset. We believe this validates our findings and explore them in more depth below.

Our most frequent self-regulated learning code was 'coding practice' (Lab 7, 58%; Lab 12, 70%). Only Falkner et al. [17] in their CS-specific self-regulated learning strategies list noted the same

Table 5: Self-regulation themes identified in our study (left hand column) and comparisons with Loksa [32], Zimmerman [61], General & CS-Specific Self-regulation strategies [17]. ">" indicates hierarchy as presented in their works.

Self-Reg Themes	Loksa	Zimmerman	General SRL strategy	CS-specific SRL strat.
Coding practice	-	-	-	Build knowledge > Practice writing code
Resources helpful	-	-	-	Build knowledge > Access resources
Self-explanation	Self-explanation	-	-	-
Goal-setting and planning	Planning	Goal-setting and planning	-	Development process > Develop design before coding
Help seeking - information	-	Seeking information	-	-
Time management	Process monitoring	Organising and transforming	-	Time management > Prioritisation
Help seeking - social	Comprehension monitoring	Seeking social assistance	Build knowledge > Talk to friends or lecturers	-
Environmental restructuring	-	-	Personal management > Reduce distractions	-
Self-evaluation	Reflection on cognition	Self-Evaluation	-	-
Motivation	-	Environmental restructuring	-	-
Resources unhelpful	-	-	-	-
Rule of thumb	-	-	-	-
Online learning - lockdown	-	-	-	-
Visualization	-	-	-	Development process > Use diagrams to describe or explain design

theme but with a much lower frequency (2%). It is clear that many of our students are convinced that the best way to learn how to code is simply to practice coding, taking a bottom-up approach, rather than a top-down approach. As the course moved to fully online learning, 12% more students noted coding practice as an important learning strategy. The discrepancy between the frequency of this theme in our study compared to Falkner et al. could be explained by the fact that we collected data from an introductory programming course while they collected data from a higher level software design course with advanced students. Such a high frequency of this code is quite understandable in an early programming course.

In our sample, 'resources helpful' is in 45% of responses in Lab 7 and 56% of responses in Lab 12. This most closely maps to Falkner's CS-specific strategy of building knowledge by accessing resources and Zimmerman's 'Seeking information' category since the students rate the resources as helpful, given the goal of finding information to accomplish homework or exam tasks. One possible explanation for the increase in this self-regulation behavior from Lab 7 to Lab 12 is as an adaptation to online learning. By the time students worked on Lab 12, they had spent many weeks in lockdown and had to rely more heavily on materials posted by the professor.

Another common theme was 'goal-setting and planning', which was one of only two of our themes that aligned with Loksa [34], Zimmerman [59], and Falkner's CS-specific strategies [17]. Although not entirely synonymous as there is room for nuance between them, Loksa's 'planning' theme included statements related to intended work goals so is considered to be synonymous with Zimmerman's 'goal setting and planning' theme. Falkner et al. [17] reported a frequency of 29% for this theme while we observed around 10% of the self-regulation responses including aspects of planning and goal setting. One explanation for this difference could be that upper division students have a better grasp on planning and goal setting due to having more experience programming in larger projects. Another explanation is that working on larger projects, as they did in Falkner's study, forced students to consider goal setting and planning to a greater extent.

Loksa's 'comprehension monitoring' is defined as "statements identifying known or unknown concepts and solutions". Our 'help seeking - social' theme was closely linked to whether a student understood a concept with most respondents noting that they only asked for help when they didn't comprehend a concept. The same strategy was noted by Falkner et al. [17] in their 'General SRL strategies' and by Zimmerman [59]. As in our analysis, this help-seeking theme was one of the less frequent strategies observed. In this study, students mentioned that working in a group reviewing concepts and seeing others code also helped them to identify gaps in their knowledge. This suggests that students surveyed at some level were monitoring their comprehension through both self-regulation and socially shared regulation.

Finally, there were many codes that did not bubble up to higher level themes simply due to a very low frequency. For instance, Falkner et al. [17] highlighted 'design' as a rarely adopted self-regulation strategy. Our results concur with their findings with zero students mentioning using this approach to supporting their individual programming practice in Lab 7 and only two mentioning it in Lab 12.

5.2 Co-regulation Themes

Unlike self-regulation, there is very little literature in computing education with which to compare our analysis of co-regulation themes (see Table 3). Instead, we will briefly discuss some of the most common trends in these themes.

In the co-regulation responses, themes such as 'resources helpful' and 'help seeking - information' were related to learning strategies within student study groups or between students. It is interesting to note the shift from reliance on coding practice and resources to more social strategies like help seeking and group learning. It was only in the self-regulation responses that we saw students commenting on resources not being helpful (less than 5% across Labs 7 and 12). Notably, the same reliance on learning resources ('resources helpful' around 50%) seen in self-regulated learning is not observed when students reflect on their strategies for learning within a group (around 10%). The same trend in frequency can be observed for 'help seeking - information' suggesting that students place less importance on resources when they work in groups.

The benefits of 'group learning', our second most frequent co-regulation theme, were noted by 30% of the students. These benefits included knowledge acquisition through comparing programming concepts and discussing code, which might utilize different algorithmic approaches from their own. The value of programming students being exposed to variations of code as a pedagogical tool is well established [36, 51]. Other respondents mention group study helps to identify and fill gaps in knowledge. Some responses also noted co-regulation strategies such as setting each other tasks, the group keeping them motivated, and making progress because of a sense of obligation to contribute to the group.

The theme of mastery and 'learning through teaching' was the third most frequent theme that emerged from our co-regulation data. The benefit of learning through teaching has been discussed in a significant body research [50]. Students who spend time teaching exhibit a better understanding and knowledge retention of the subject matter than students who do not teach others [10]. Notably, this theme did not appear in the self-regulation responses. The closest self-regulation theme is that of 'self-explanation' where students reinterpreted the course materials, for example, by making their own notes or adding comments to example code.

The students in our study appear to place more importance on goal setting, planning and time management when reflecting on their self-regulation (average 10%) than when working in groups (average 5%). However, it is interesting that most co-regulation responses that mentioned this theme noted the positive influence that working in a group has on their planning and time management.

An average of 5% of respondents to the co-regulation prompt were explicitly group averse, highlighting the perceived downsides to working in a group such as difficulty focusing, working at a pace that suits the group not the individual, and pressure to perform. Finally, despite the fact that the quantitative analysis showed that students who engaged more in co-regulation performed more poorly on the C-project and final exam, over half of the student reflective responses analyzed spoke to a perceived benefit in group learning and social help seeking.

5.3 Connecting with Hadwin's SSRL model

Hadwin et al. [24] describe group work as unfolding over four loosely connected phases that can be visited and revisited in the same session. Each session of group work builds on the last, cycling through the SSRL phases in new ways. Each phase, therefore, evolves over the course of the group's study interactions.

In Phase 1, groups work together to understand the task. Before work on the programming assignment can begin, members of the group each identify and interpret the task and then compare that understanding to that of their peers. Co-regulation themes we identified that fit this phase include 'help seeking - social' and 'help seeking - information'. The following quotes from students illustrate this behavior:

- "I work with my friends when I become stuck or if they become stuck and we help each other to understand the problem better and tackle it."
- "The main focus we try to do when studying in groups is to bring to light any misconceptions we have about the tasks we are meant to complete. It works best as it tests the understanding, if we ourselves are able to teach the subject to someone who is struggling."

During Phase 2, with a shared understanding of the task at hand, groups negotiate their plans to reach their shared goal(s) and the standard by which they will evaluate the finished result. We identified several co-regulation themes that fit into this phase, including 'group planning' and 'help seeking - time management'. The following quotes from students illustrate the behaviors in phase:

- "For the times I do study in groups, I try set goals and tell them to other people of what I will do each session and get people to plan out goals for each session."
- "When learning together, I think it is important to be collaborative and plan things together."
- "When studying with a group/peer for this course, we often set out a plan and goals for each study session and work our way through to those goals. Whether it be watching lectures or doing exercises, we work through any queries to meet the end goal."

Of course, not everyone followed this pattern, as one student wrote: "We don't really direct the session, everyone does what they have to do. If one has a question, he can always ask the others for a quick response and explanation."

In Phase 3, groups work together on the task, collaboratively utilizing multiple strategies to achieve the goal. These strategies are constructed by the group itself and can include cognitive, metacognitive, socio-emotional, and motivational strategies. Not each member of the group necessarily needs to utilize every strategy employed by the group. Rather, individuals with those strengths will utilize them for the good of the group and its ultimate success. Co-regulation themes that we identified that fit this phase include 'socially shared regulation' and 'motivation'. Representative quotes from students below illustrate these ideas:

- "Sometimes I have been watching lectures with friends to motivate each other and if there are times where I don't fully understand an explanation or want to solidify my understanding, a short discussion may occur."
- "I ensure that we stay on task and do not deviate from the matter."
- "We try to encourage one another to keep on track and reach the finish line together."
- "As a group we do not rely on one person to support everyone, but we all share our knowledge of this course. This is really helpful as it improves my knowledge in concepts I already understand, while at the same time understand the explanations of the parts I am confused about."

Phase 4 is where groups make small- and large-scale adaptations to their plans, standards, and goals. This could range from a small optimization to the current learning sub-task to a large-scale pivot to an entirely different plan or goal. The co-regulation theme that illustrates this phase is 'group learning' and 'socially shared regulation'. The following quotes from students illustrate this phase:

- "To find what works best for the group, everyone needs to feedback and give ideas on what they find useful for their own learning in hopes to incorporate it into the group learning."
- "We found it useful to step away from the computer and break down the problem on paper and work out what we were actually trying to solve, usually with pseudo code."
- "If we were stuck on a problem, we presented it to the group. It was interesting to see how others would approach the same question and I believe that that greatly improved my ability to adapt my thinking."

The third most frequent co-regulation theme that we identified, 'learn through teaching', appeared in every phase. This highlights the versatility of some of our themes, which could fit into multiple phases of co-regulation within groups.

5.4 Limitations

The primary limitation of this work is that students went from in-class to entirely online very close to when the C module began. This means that group study behaviors developed during the MATLAB portion of the course would need to be altered at the least and could have been completely disrupted at the worst. As discussed above, we explored how performance in the course compared to previous iterations. We found that the data for 2021 is much more similar to pre-pandemic versions of the course than it is to the 2020 version, but no such comparisons are available for the qualitative data describing self- and co-regulation. The NLP-generated counts of metacognitive expressions we used to compare our qualitative subsample to the whole sample may be one way of at least comparing metacognition between time periods. However, the NLP method does not measure phrases related to regulating others' cognition (e.g., third person phrases) that might be valuable additions for validating co-regulation across datasets in particular. Even given this limitation, we believe the NLP tool is the best validated approach to get a broad sense of student metacognition in written responses over a large dataset.

6 CONCLUSION

We presented a mixed-methods study examining student group study behavior through the lens of the Socially Shared Regulated Learning (SSRL) model by Hadwin et al. [24]. We now return to our research questions. In response to RQ1, *How do students express their learning strategies both when working alone and when working in groups, and how do these align with existing models of self-regulation and co-regulation?*, we found rich and robust descriptions of metacognition and group-oriented regulation in students who studied with peers. Furthermore, written responses by students regarding naturally occurring group studying habits in an introductory programming course aligned well with the SSRL model. We provided the first validation of the model in a programming context and the first elicitation of computing-specific co-regulation themes. We situate these contributions in the context of related work in programming education in Table 5. In response to RQ2, *To what extent do written expressions of self-regulation, co-regulation, and socially shared regulation relate to student performance?*, we found a slight positive correlation with increased self-regulation and performance and a slight negative correlation with increased co-regulation and performance. Despite this surprising finding, over half of the co-regulation responses that we analyzed mentioned perceived positive benefits.

This exploratory research is at the frontier of self-regulation and co-regulation in introductory programming and we have identified a number of relevant avenues for future work. First, the negative correlation we found between co-regulation and performance should be investigated. If confirmed, what implications does this have for the future of computing education, especially in the context of online learning environments? Group work is obviously an important skill to learn in computing, so can this negative correlation be mitigated or even reversed? Second, we have identified co-regulation themes that can now be leveraged by pedagogical interventions in introductory programming courses, such as group work designed to help students learn through teaching. Finally, these themes can be utilized to understand observations of group study behavior. How do groups move through Hadwin's phases? When they do, which themes appear and in what order? Do new themes appear when watching a group study together as opposed to asking students to reflect on their group study practices? There are many more questions and avenues of future work. We invite the community to use this work as a starting point to explore how group work mechanisms and behaviors impact novice programmer metacognition and ultimately student learning.

ACKNOWLEDGMENTS

This work is funded in part by the National Science Foundation under grant #1941642.

REFERENCES

[1] Eshwar Bachu and Margaret Bernard. 2012. A Computer Supported Collaborative Learning (CSCL) Model for Educational Multiplayer Games. In *11th International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government*. Las Vegas, NV.

[2] Albert Bandura. 1986. The Explanatory and Predictive Scope of Self-efficacy Theory. *Journal of Social and Clinical Psychology* 4, 3 (1986), 359–373.

[3] Brett A. Becker and Keith Quille. 2019. 50 Years of CS1 at SIGCSE: A Review of the Evolution of Introductory Programming Education Research. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. ACM, New York, NY, USA, 338–344. https://doi.org/10.1145/3287324.3287432

[4] Susan Bergin, Ronan Reilly, and Desmond Traynor. 2005. Examining the Role of Self-Regulated Learning on Introductory Programming Performance. In *Proceedings of the First International Workshop on Computing Education Research* (Seattle, WA, USA) *(ICER '05)*. ACM, New York, NY, USA, 81–86. https://doi.org/10.1145/1089786.1089794

[5] Margaret Bernard and Eshwar Bachu. 2015. Enhancing the Metacognitive Skill of Novice Programmers through collaborative learning. In *Metacognition: Fundaments, applications, and trends*. Springer, Cham, CH, 277–298.

[6] Katerine Bielaczyc, Peter L. Pirolli, and Ann L. Brown. 1995. Training in Self-Explanation and Self-Regulation Strategies: Investigating the Effects of Knowledge Acquisition Activities on Problem Solving. *Cognition and Instruction* 13, 2 (1995), 221–252. https://doi.org/10.1207/s1532690xci1302_3

[7] Monique Boekaerts and Lyn Corno. 2005. Self-Regulation in the Classroom: A Perspective on Assessment and Intervention. *Applied Psychology* 54, 2 (2005), 199–231. https://doi.org/10.1111/j.1464-0597.2005.00205.x

[8] Nigel Bosch, Yingbin Zhang, Luc Paquette, Ryan Baker, Jaclyn Ocumpaugh, and Gautam Biswas. 2021. Students' Verbalized Metacognition during Computerized Learning. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–12.

[9] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. https://doi.org/10.1191/1478088706qp063oa

[10] Peter A Cohen, James A. Kulik, and Chen-Lin C. Kulik. 1982. Educational Outcomes of Tutoring: A Meta-analysis of Findings. *American Educational Research Journal* 19, 2 (1982), 237–248. https://doi.org/10.3102/00028312019002237

[11] Paul Denny, Brett A. Becker, Nigel Bosch, James Prather, Brent Reeves, and Jacqueline Whalley. 2022. Novice Reflections During the Transition to a New Programming Language. In *Proceedings of the 53rd ACM Technical Symposium V.1 on Computer Science Education* (Providence, RI, USA) *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 948–954. https://doi.org/10.1145/3478431.3499314

[12] Paul Denny, John Hamer, Andrew Luxton-Reilly, and Helen Purchase. 2008. PeerWise: Students Sharing Their Multiple Choice Questions. In *Proceedings of the Fourth International Workshop on Computing Education Research*. Association for Computing Machinery, New York, NY, 51–58.

[13] Paul Denny, James Prather, Brett A. Becker, Zachary Albrecht, Dastyni Loksa, and Raymond Pettit. 2019. A Closer Look at Metacognitive Scaffolding: Solving Test Cases Before Programming. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '19)*. Association for Computing Machinery, New York, NY, USA, Article 11, 10 pages. https://doi.org/10.1145/3364510.3366170

[14] Paul Denny, Jacqueline Whalley, and Juho Leinonen. 2021. Promoting Early Engagement with Programming Assignments Using Scheduled Automated Feedback. In *Australasian Computing Education Conference*. Association for Computing Machinery, New York, NY, 88–95.

[15] Anastasia Efklides. 2008. Metacognition: Defining its Facets and Levels of Functioning in Relation to Self-regulation and Co-regulation. *European Psychologist* 13, 4 (2008), 277–287.

[16] MD Evans. 1996. A New Emphasis & Pedagogy for a CS1 Course. *ACM SIGCSE Bulletin* 28, 3 (1996), 12–16.

[17] Katrina Falkner, Rebecca Vivian, and Nickolas J.G. Falkner. 2014. Identifying Computer Science Self-Regulated Learning Strategies. In *Proceedings of the 2014 Conference on Innovation and Technology in Computer Science Education* (Uppsala, Sweden) *(ITiCSE '14)*. ACM, New York, NY, USA, 291–296. https://doi.org/10.1145/2591708.2591715

[18] Kasper Fisker, Davin McCall, Michael Kölling, and Bruce Quig. 2008. Group Work Support for the BlueJ IDE. In *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education*. Association for Computing Machinery, New York, NY, 163–168.

[19] John H Flavell. 1979. Metacognition and Cognitive Monitoring: A New Area of Cognitive-Developmental Inquiry. *American Psychologist* 34, 10 (1979), 906.

[20] D Randy Garrison and Zehra Akyol. 2015. Thinking Collaboratively in Educational Environments: Shared Metacognition and Co-regulation in Communities of Inquiry. In *Educational developments, practices and effectiveness*. Springer, Cham, CH, 39–52.

[21] Ann Q Gates. 1996. Integrating a Problem-solving Methodology and Group Skills Into CS1. In *Proceedings of 9th Conference on Software Engineering Education*. IEEE, Piscataway, NJ, 6–15.

[22] Lisa Given. 2016. *100 Questions (and Answers) About Qualitative Research*. Vol. 68. SAGE Publications, Thousand Oaks, CA.

[23] Graciela Gonzalez. 2006. A Systematic Approach to Active and Cooperative Learning in CS1 and its Effects on CS2. In *Proceedings of the 37th SIGCSE technical symposium on Computer science education*. Association for Computing Machinery, New York, NY, 133–137.

[24] Allyson Fiona Hadwin, Sanna Järvelä, and Mariel Miller. 2018. Self-regulated, Co-regulated, and Socially Shared Regulation of Learning. In *Handbook of Self-regulation of Learning and Performance, 2nd Edition*, D. H. Schunk and J. A. Greene Greene (Eds.). Vol. 30. Routledge/Taylor & Francis Group, New York, NY, 83–106.

[25] Eddie Huang, Hannah Valdiviejas, and Nigel Bosch. 2019. I'm sure! Automatic detection of metacognition in online course discussion forums. In *Proceedings of the 8th International Conference on Affective Computing and Intelligent Interaction (ACII 2019)*. IEEE, Piscataway, NJ, 241–247. https://doi.org/10.1109/ACII.2019.8925506

[26] Tarja-Riitta Hurme, Sanna Järvelä, Kaarina Merenluoto, and Pekka Salonen. 2015. What Makes Metacognition as Socially Shared in Mathematical Problem Solving? In *Metacognition: Fundaments, Applications, and Trends*. Springer, Cham, CH, 259–276.

[27] Kalle Ilves, Juho Leinonen, and Arto Hellas. 2018. Supporting Self-Regulated Learning with Visualizations in Online Learning Environments. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (Baltimore, Maryland, USA) *(SIGCSE '18)*. Association for Computing Machinery, New York, NY, USA, 257–262. https://doi.org/10.1145/3159450.3159509

[28] Einari Kurvinen, Rolf Lindén, Teemu Rajala, Erkki Kaila, Mikko-Jussi Laakso, and Tapio Salakoski. 2012. Computer-Assisted Learning in Primary School Mathematics Using ViLLE Education Tool. In *Proceedings of the 12th Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '12)*. Association for Computing Machinery, New York, NY, USA, 39–46. https://doi.org/10.1145/2401796.2401801

[29] Noel LeJeune. 2003. Critical Components for Successful Collaborative Learning in CS1. *Journal of Computing Sciences in Colleges* 19, 1 (2003), 275–285.

[30] Richard Lobb and Jenny Harlow. 2016. Coderunner: A Tool for Assessing Computer Programming Skills. *ACM Inroads* 7, 1 (feb 2016), 47–51. https://doi.org/10.1145/2810041

[31] Dastyni Loksa and Amy J. Ko. 2016. The Role of Self-Regulation in Programming Problem Solving Process and Success. In *Proceedings of the 2016 ACM Conference on International Computing Education Research* (Melbourne, VIC, Australia) *(ICER '16)*. Association for Computing Machinery, New York, NY, USA, 83–91. https://doi.org/10.1145/2960310.2960334

[32] Dastyni Loksa, Amy J. Ko, Will Jernigan, Alannah Oleson, Christopher J. Mendez, and Margaret M. Burnett. 2016. Programming, Problem Solving, and Self-Awareness: Effects of Explicit Guidance. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1449–1461. https://doi.org/10.1145/2858036.2858252

[33] Dastyni Loksa, Lauren Margulieux, Brett A. Becker, Michelle Craig, Paul Denny, Raymond Pettit, and James Prather. 2021. Metacognition and Self-Regulation in Programming Education: Theories and Exemplars of Use. *ACM Trans. Comput. Educ.* (Dec 2021). https://doi.org/10.1145/3487050 Just Accepted.

[34] Dastyni Loksa, Benjamin Xie, Harrison Kwik, and Amy J. Ko. 2020. Investigating Novices' In Situ Reflections on Their Programming Process. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (Portland, OR, USA) *(SIGCSE '20)*. Association for Computing Machinery, New York, NY, USA, 149–155. https://doi.org/10.1145/3328778.3366846

[35] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A. Becker, Michail Giannakos, Amruth N. Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and et al. 2018. Introductory Programming: A Systematic Literature Review. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (Larnaca, Cyprus) *(ITiCSE 2018 Companion)*. ACM, New York, NY, USA, 55–106. https://doi.org/10.1145/3293881.3295779

[36] Dara Mahdid, Piotr Szybek, and Führer. 2015. A Study on Variation Technique in Courses on Scientific Computing. *Science Journal of Education* 3, 3 (2015), 60–67. https://doi.org/10.11648/j.sjedu.20150303.13

[37] Katerina Mangaroska, Kshitij Sharma, Dragan Gasevic, and Michail Giannakos. 2022. Exploring students' cognitive and affective states during problem solving through multimodal data: Lessons learned from a programming activity. *Journal of Computer Assisted Learning* 38, 1 (2022), 40–59. https://doi.org/10.1111/jcal.

12590 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/jcal.12590

[38] Murali Mani and Quamrul Mazumder. 2013. Incorporating Metacognition into Learning. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education* (Denver, Colorado, USA) *(SIGCSE '13)*. Association for Computing Machinery, New York, NY, USA, 53–58. https://doi.org/10.1145/2445196.2445218

[39] Laurie Murphy and Josh Tenenberg. 2005. Do Computer Science Students Know What They Know? A Calibration Study of Data Structure Knowledge. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (Caparica, Portugal) *(ITiCSE '05)*. Association for Computing Machinery, New York, NY, USA, 148–152. https://doi.org/10.1145/1067445.1067488

[40] Paul R. Pintrich. 2000. Chapter 14 - The Role of Goal Orientation in Self-Regulated Learning. In *Handbook of Self-Regulation*, Monique Boekaerts, Paul R. Pintrich, and Moshe Zeidner (Eds.). Academic Press, San Diego, 451 – 502. https://doi.org/10.1016/B978-012109890-2/50043-3

[41] Paul R Pintrich and Elisabeth V De Groot. 1990. Motivational and Self-regulated Learning Components of Classroom Academic Performance. *Journal of Educational Psychology* 82, 1 (1990), 33–40.

[42] Paul R. Pintrich, David A. F. Smith, Teresa Garcia, and Wilbert J. Mckeachie. 1993. Reliability and Predictive Validity of the Motivated Strategies for Learning Questionnaire (MSLQ). *Educational and Psychological Measurement* 53, 3 (1993), 801–813. https://doi.org/10.1177/0013164493053003024

[43] James Prather, Brett A. Becker, Michelle Craig, Paul Denny, Dastyni Loksa, and Lauren Margulieux. 2020. What Do We Think We Think We Are Doing? Metacognition and Self-Regulation in Programming. In *Proceedings of the 2020 ACM Conference on International Computing Education Research* (Virtual Event, New Zealand) *(ICER '20)*. Association for Computing Machinery, New York, NY, USA, 2–13. https://doi.org/10.1145/3372782.3406263

[44] James Prather, Raymond Pettit, Brett A. Becker, Paul Denny, Dastyni Loksa, Alani Peters, Zachary Albrecht, and Krista Masci. 2019. First Things First: Providing Metacognitive Scaffolding for Interpreting Problem Prompts. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 531–537. https://doi.org/10.1145/3287324.3287374

[45] James Prather, Raymond Pettit, Kayla McMurry, Alani Peters, John Homer, and Maxine Cohen. 2018. Metacognitive Difficulties Faced by Novice Programmers in Automated Assessment Tools. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (Espoo, Finland) *(ICER '18)*. Association for Computing Machinery, New York, NY, USA, 41–50. https://doi.org/10.1145/3230977.3230981

[46] James Prather, Raymond Pettit, Kayla Holcomb McMurry, Alani Peters, John Homer, Nevan Simone, and Maxine Cohen. 2017. On Novices' Interaction with Compiler Error Messages: A Human Factors Approach. In *Proceedings of the 2017 ACM Conference on International Computing Education Research* (Tacoma, Washington, USA) *(ICER '17)*. Association for Computing Machinery, New York, NY, USA, 74–82. https://doi.org/10.1145/3105726.3106169

[47] Leonardo Silva. 2021. Fostering Programming Students Regulation of Learning Using a Computer-Based Learning Environment. In *2021 International Symposium on Computers in Education (SIIE)*. IEEE, Piscataway, NJ, 1–5.

[48] Leonardo Silva, António José Mendes, Anabela Gomes, and Gabriel Fortes Cavalcanti de Macêdo. 2021. Regulation of Learning Interventions in Programming Education: A Systematic Literature Review and Guideline Proposition. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (Virtual Event, USA) *(SIGCSE '21)*. Association for Computing Machinery, New York, NY, USA, 647–653. https://doi.org/10.1145/3408877.3432363

[49] Leonardo S Silva. 2020. Investigating the Socially Shared Regulation of Learning in the Context of Programming Education. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*. Association for Computing Machinery, New York, NY, 575–576.

[50] Martin Stigmar. 2016. Peer-to-peer Teaching in Higher Education: A Critical Literature Review. *Mentoring & Tutoring: Partnership in Learning* 24, 2 (2016), 124–136. https://doi.org/10.1080/13611267.2016.1178963 arXiv:https://doi.org/10.1080/13611267.2016.1178963

[51] Jarkko Suhonen, Janet Davies, Errol Thompson, and Kinshuk. 2007. Applications of Variation Theory in Computing Education. In *Proceedings of the Seventh Baltic Sea Conference on Computing Education Research - Volume 88* (Koli National Park, Finland) *(Koli Calling '07)*. Australian Computer Society, Inc., AUS, 217–220.

[52] Tammy VanDeGrift, Tamara Caruso, Natalie Hill, and Beth Simon. 2011. Experience Report: Getting Novice Programmers to THINK about Improving Their Software Development Process. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (Dallas, TX, USA) *(SIGCSE '11)*. Association for Computing Machinery, New York, NY, USA, 493–498. https://doi.org/10.1145/1953163.1953307

[53] Arto Vihavainen, Craig S. Miller, and Amber Settle. 2015. Benefits of Self-Explanation in Introductory Programming. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (Kansas City, Missouri, USA) *(SIGCSE '15)*. Association for Computing Machinery, New York, NY, USA, 284–289. https://doi.org/10.1145/2676723.2677260

[54] Henry M Walker. 1997. Collaborative Learning: A Case Study for CS1 at Grinnell College and Austin. In *Proceedings of the Twenty-Eighth SIGCSE Technical Symposium on Computer Science Education*. Association for Computing Machinery, New York, NY, 209–213.

[55] Philip H Winne and Allyson F Hadwin. 1998. Studying as Self-Regulated Engagement in Learning. In *Metacognition in Educational Theory and Practice*. Routledge, New York, NY, 277–304.

[56] Aman Yadav, Chris Mayfield, Sukanya Kannan Moudgalya, Clif Kussmaul, and Helen H. Hu. 2021. Collaborative Learning, Self-Efficacy, and Student Performance in CS1 POGIL. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (Virtual Event, USA) *(SIGCSE '21)*. Association for Computing Machinery, New York, NY, USA, 775–781. https://doi.org/10.1145/3408877.3432373

[57] Aman Yadav, Ceren Ocak, and Amber Oliver. 2022. Computational Thinking and Metacognition. *TechTrends* (2022), 1–7.

[58] Lanqin Zheng, Xin Li, Xuan Zhang, and Wei Sun. 2019. The Effects of Group Metacognitive Scaffolding on Group Metacognitive Behaviors, Group Performance, and Cognitive Load in Computer-supported Collaborative Learning. *The Internet and Higher Education* 42 (2019), 13–24.

[59] Barry J Zimmerman. 1989. A Social Cognitive View of Self-regulated Academic Learning. *Journal of Educational Psychology* 81, 3 (1989), 329.

[60] Barry J. Zimmerman. 2000. Chapter 2 - Attaining Self-Regulation: A Social Cognitive Perspective. In *Handbook of Self-Regulation*, Monique Boekaerts, Paul R. Pintrich, and Moshe Zeidner (Eds.). Academic Press, San Diego, 13 – 39. https://doi.org/10.1016/B978-012109890-2/50031-7

[61] Barry J. Zimmerman and Manuel Martinez Pons. 1986. Development of a Structured Interview for Assessing Student Use of Self-Regulated Learning Strategies. *American Educational Research Journal* 23, 4 (1986), 614–628. https://doi.org/10.3102/00028312023004614