How Can We Know When We See It? A Systematic Review of Cognitive Control Skills and Behaviors

MOLLY DOMINO, ALEXANDRA THOMPSON, ALEXANDER HICKS, ALAN JAMIESON, KHUSHI PARAJULI, BOB EDMISON, STEPHEN H. EDWARDS, and CLIFFORD A. SHAFFER, Virginia Tech, USA

Academic success requires not only knowing what to learn, but also how to learn it. Cognitive control, an umbrella term for metacognition and self-regulation, is critical to the learning process, allowing a student to exercise control over their thoughts and behaviors. There is growing interest within the Computer Science Education community to better understand how these constructs are used when learning to program. However, while these skills carry a direct link with academic success, they are highly internal and therefore often difficult to observe. The primary goal of this work is to help CS educators and researchers recognize when a student is engaging in cognitive control by documenting what skills the research community has identified as important, and what behaviors can be observed to indicate those skills are present.

This work presents a rapid review of research papers studying cognitive control in post-secondary computer science education. A corpus of 51 works from the ACM Full Guide to Computing Literature online library was constructed and the authors identified 11 cognitive control skills and 10 behavioral indicators of a skill in action from these works. Findings from this review indicate that future work should diversify not only what skills we study but how we approach that study. We highlight two skills, *Monitoring Correctness* and *Scheduling* within the work reviewed here as they have the greatest body of evidence linking them to academic success. Two more skills, *Task Analysis* and *Decomposing* also appear to be a promising direction for future work. We also call out other skills that the corpus indicates as promising directions for future work.

CCS Concepts: • **Social and professional topics → Computer science education**.

Additional Key Words and Phrases: Computer Science Education, Metacognition, Self-Regulation

1 Introduction

Metacognition and self-regulation are critical to the learning process, allowing a student to manage their thoughts and behaviors to meet their learning goals. Collectively called "cognitive control", there is growing interest within the Computer Science Education community to better understand how these constructs are used when learning to program. Many cognitive control skills have a demonstrated link with academic success [23, 50, 54, 84], and studies show students benefit from explicit instruction on their use [53, 61].

However while they are highly valuable, cognitive control skills also present some challenges to educators and researchers. They are highly internal, making it difficult to observe or measure the impact of interventions. Within the context of CS, this is compounded by the very nature of the field. Major assessment activities are often done by an individual in private. Furthermore, as Prather et al. point out, "computing has unique characteristics that make learning and teaching particular topics arguably different than in other disciplines" [75]. Loksa et al.

Authors' addresses: Molly Domino, mollydomino@vt.edu; Alexandra Thompson, alexthompson06@vt.edu; Alexander Hicks, awh4kc@vt.edu; Alan Jamieson, a.jamieson@northeastern.edu; Khushi Parajuli, kparajuli@vt.edu; Bob Edmison, bedmison@vt.edu; Stephen H. Edwards, edwards@cs.vt.edu; Clifford A. Shaffer, shaffer@vt.edu, Virginia Tech, Blacksburg, VA, USA.

ACM 1946-6226/2026/7-ART

https://doi.org/10.1145/3821524

go on to posit : “entirely new theories of metacognition and Self-Regulated Learning (SRL) might arise from their application in this context [programming]” [52]. Thus, it is beneficial to the Computing Education Research (CER) community to know what skills are most valuable to target. Identifying ways of observing or measuring those skills is also critical as this allows us to determine if an intervention has had the desired impact.

To start addressing the need, this review seeks to answer two research questions:

- **RQ1:** What cognitive-control skills have been examined in post-secondary computer science education research, and what behaviors have been used to indicate skills?
- **RQ2:** Which cognitive-control skills (or associated behaviors) have a demonstrated relationship with post-secondary student success in computer science education?

We present a Rapid Review of cognitive control within the computing education literature. Our methods use the framework laid out by Plüddemann et al. [73]. The Rapid Review approach adjusts a systematic review framework to a more limited reviewer pool during study selection and extraction while still seeking to mitigate bias at every step.

Our work leads us to recommend that future research would benefit from diversifying what is studied and what methodologies are used in the process. We note that while a diverse set of skills are recognized as important, works within this corpus tend to focus attention on a subset of these skills. These works also tend to approach the examination of these skills in the same, specific ways.

While examining how cognitive control works in other settings is critical, we focus this work on post-secondary students learning Computer Science in class, rather than K-12 students or professionals engaged in workplace learning. Practically, this choice was made to keep the scope of our search focused as cognitive control, even within this limited scope, our initial search yielded an extensive corpus to filter through. Additionally, direct comparisons between K-12 students, post-secondary students, and workplace learners may not be wholly appropriate given the wide developmental differences between these three groups.

The paper is organized as follows. Section 2 defines the terms self-regulation and metacognition for the purpose of this paper. We then give a brief theoretical history of both constructs and how those theories contribute to our review. Section 3 details the process used to create the corpus of papers reviewed here as well as how data was extracted and synthesized. This section also discussed some potential areas of bias within that process. We found eleven skills were discussed in this body of papers. These skills span what a student does before working, while working, and after working. Section 4 provides a brief overview of our interpretation of the skills, how they are typically defined, and what indicators are commonly used to observe them. Section 5 discusses trends that we observed related to study of these skills. Section 6 focuses on skills from Section 4 that have been shown to relate to academic success. We also discuss two skills (*Task Analysis* and *Decomposing*) that look promising, but did not see the same quantity of research in this corpus. Section 7 further analyzes what skills have been well-studied in this corpus and what methodologies have been used to relate them to academic success. This section also highlights some overall observations on methodologies used to study cognitive control among the studies in this corpus and presents recommendations for future work. Section 8 presents our conclusions.

2 Background

2.1 Clarifying Terms

Literature on self-regulation and metacognition often uses these terms to convey the same concepts. Overall, “metacognition” is more associated with psychology, while “self-regulation” is more associated with education [75]. No universal or clear-cut boundary between these terms exists. Self regulation was originally considered one component of metacognition before developing into an independent construct. Thus, it is in some ways a subdomain of metacognition. Yet “metacognitive monitoring” is an aspect of many major theoretical frameworks

of SRL, making metacognition a subdomain of self regulation as well. Terms like "metacognitive self-regulation" also appear in the literature, further confusing things. Each paper in our review defined these terms for themselves.

Since these words can mean different things to different people, Prather et al. and Loksa et al. [52, 75] have previously recommended that researchers create their own working definitions. Following that recommendation, we now define how we use these terms within this work.

- **Metacognition** concerns the knowledge generated from the process of "thinking about thinking".
- **Self-Regulation** concerns the application of metacognitive knowledge to manage behavior in order to complete a task.
- **Cognitive Control** is an umbrella term to cover both metacognition and self-regulation.

These definitions are also supported by those previous works [52, 75]. Loksa et al. [52] differentiate these terms more thoroughly and their review is a good resource for readers interested in a deeper exploration of these concepts.

2.2 Theoretical Models of of Metacognition

Theories of metacognition often focus on the relationship between knowledge of oneself from regulation of cognition. In the book 'Metacognition in Learning and instruction: Theory, research and practice', the authors note that: "Most researchers make a distinction between two components of metacognition, *knowledge of cognition* and *regulation of cognition*" [36]. The former describe learners' declarative knowledge of how they think, procedural knowledge of strategies to learn most effectively, and conditional knowledge of when to apply those strategies. The latter describes an application of knowledge to regulate their learning. This is somewhat different to the working definition of metacognition from the previous section. For the purposes of this paper, only the types of self-knowledge a learner needs to succeed are within the scope of metacognition. Application of knowledge fits more accurately within our own definition of self-regulation.

This section will discuss the historical context into how metacognition has been conceptualized and the ways in which knowledge of cognition and regulation of cognition have become more distinct over time. This additional context should also serve to underline why we opted to focus our working definition of metacognition on knowledge rather than application.

Flavell [25] is considered a seminal researcher into theories of metacognition. His model breaks metacognition into four sections: *metacognitive knowledge*, *metacognitive experiences*, *goals*, and *actions*. *Metacognitive knowledge* is the knowledge that impacts cognitive areas. For example, "a child's acquired belief that, unlike many of her friends, she is better at arithmetic than at spelling" [25]. *Metacognitive experiences* are cognitive or affective responses like suddenly feeling lost in a lecture. *Goals* (or tasks) refer to the "objectives of a cognitive enterprise" [25]. *Actions* (or strategies) are the things one thinks and does to achieve one's goals. We do see a separation of knowledge and regulation within Flavell's framework. For example, 'Metacognitive Knowledge' and indeed 'metacognitive experiences' clearly focus on forms 'knowledge of cognition'. However, the other two categories do not as neatly fit into either knowledge or regulation. 'Actions' arguably fits under the category of 'regulation of cognition', but also implicitly requires a student has the procedural knowledge of good learning strategies. Furthermore, one could make the argument that 'goals' are neither knowledge nor regulation of cognition. Thus, this seminal framework makes a start at distinguishing between knowledge of cognition and regulation of learning, but does not fully commit to the idea.

Over the next few decades, we see a shift towards framing metacognition as a type of intelligence. In 1985 Sternberg et al. synthesized these four components in their "Triarchic theory of Human Intelligence" [93]. One aspect of this framework is the "componential subtheory" that encompasses all "mental mechanisms underlying intelligent performance". This subtheory includes a component of higher-order decision-making to plan and solve problems.

Similarly, the 2001 revision of Bloom's 1956 taxonomy of cognition also adds 'metacognitive knowledge' as a new aspect of knowledge. This metacognition category contained three subcategories: "strategic knowledge", "knowledge about cognitive tasks", and "self-knowledge". *Strategic Knowledge* covers a knowledge of learning strategies. *Knowledge about cognitive tasks* covers a learner's assessment of their surroundings and the conditional knowledge of which learning strategies are most appropriate for the setting. *Self-Knowledge* covers a learner's declarative knowledge of their own mind. Here, we see a much greater focus on knowledge of cognition. These forms of knowledge are even broken down in the revised Bloom's Taxonomy to demonstrate some separate ways metacognitive knowledge can manifest. It is also continually separated from regulation of cognition.

Thus, the separation of knowledge from application from these later formulations also helped inform our own working definitions described in Section 2.1.

2.3 Theories of Self-Regulation

Self-regulation in the context of learning has many frameworks, each tying in other educational constructs. In their 2017 meta-analysis, Panadero et al. identify six major approaches to self-regulation theory [63], as represented by the following papers: Zimmerman [106]; Pintrich [72]; Winne and Hadwin [97]; Efklides [20]; Hadwin, Järvelä, and Miller [32]; and Boekaerts [4]. This section will focus on the first three models since the other three were not well discussed in our corpus of works.

Zimmerman's Cyclical Phase model [106], developed in 1986, remains one of the most well-recognized conceptions of self-regulation with respect to learning. Stemming from his work with Bandura, the Cyclical Phase model divides the process into three distinct phases that a student moves through sequentially as they complete work.

The *forethought* phase comes before starting, when a student prepares to do a task. The task is analyzed, goals are set, and a student might develop a plan on how to approach their work. A student might also consider some aspects of motivation like their self-efficacy, expectations, interest, and personal goals for their work. Once a student begins to work, they enter the *performance* phase. During this phase, students might observe themselves and make adjustments to better suit their needs. These adjustments could be strategic like keeping track of time or seeking help when stuck, or they could be motivational like creating an incentive for finishing. After a student has finished their work, they progress to the *self-reflection* phase. Here, students look back on the work they accomplished and consider the variables that might have contributed to the quality of the work they produced. This evaluation also tends to have an emotional component as well. This self-reflection and the emotional responses of that reflection feed into the forethought phase of the next task a student does (which is why the model is called cyclical). For example, if a student performed below their own expectations, that might impact their self-efficacy considerations in their next *forethought* phase.

Pintrich's framework for SRL is commonly referenced as an influence in this corpus. While similar to Zimmerman's, this model puts an even greater emphasis on how the self-regulatory process and motivation interact [72]. Pintrich's model breaks self-regulation into two dimensions: four stages of the process of self-regulation and four areas that one can regulate. The four phases are: *Forethought, planning, and activation*; *Monitoring*; *Control*; and *Reaction and Reflection.* At each stage, a student can regulate their cognition, motivation/affect, behavior, and context. For example, in the *Forethought, planning, and activation* phase a student might regulate their cognition by setting a goal. They might regulate their behavior by setting aside time to work on the task. 'Context' represents the variables surrounding the student and the task like cultural context or physical environment. While this first phase does not have a contextual dimension, the *Control* phase can see students changing contexts, like going to the library for a quieter place to study [72].

Pintrich's model serves as the foundation for the Motivated Strategies for Learning Questionnaire (MSLQ) that is used in eight of the studies discussed in this work. While two studies used the Student Perceptions

of Classroom Knowledge Building (SPOCK) [24, 68] and a number of other studies sometimes created their own custom surveys, the MSLQ was, by far, the most discussed inventory within this corpus. Previous reviews both within CS education research [52], and in more general education research [83] confirm the popularity of this inventory across domains. The MSLQ inventory consists of 15 sub-scales measuring different aspects of Self-Regulated Learning. While one sub-scale directly measures Metacognitive Self-Regulation, other constructs are also measured: self-efficacy, test anxiety, and goal orientation. All 81 questions in the MSLQ are phrased as statements that students must rate on a 7-point Likert-scale from "not at all true of me" to "very true of me" [69].

The Winne-Hadwin model has less focus on motivation, instead focusing on metacognitive and cognitive strategy use [97]. Like Pintrich's model, Winne and Hadwin include both a set of self-regulation stages and a set of variables those stages interact with. The phases for this model are: *task definition*, *goal setting and planning*, *enacting study tactics and strategies*, and *metacognitively adapting studying*. The other dimension is described through the "COPES" mnemonic device: **C**onditions describe "features a learner perceives to influence their work" [99] both internal and external to the learner. **O**perations are the ways a student can process information. **P**roducts are the physical artifacts created by doing a task or assignment. **E**valuations represent how one assesses the value of the work in accordance with their **S**tandards, which represents the internal rubric one uses when conducting those evaluations. Unlike the models of Zimmerman or Pintrich, the Winne-Hadwin model does not represent a generally time-ordered sequence of phases a student might go through. Instead, students flow between phases in a feedback loop as they self-regulate.

While Zimmerman's, Pintrich's, and Winne and Hadwin's models have received the most attention in the studies reviewed here, there are three other notable models. Boekaerts [5] "Dual Processing Model" focuses more on how goals and specific situations influence self-regulation. While Winne began to focus on data-driven models of studying SRL [99], Hadwin has since collaborated with Järvelä and Miller to focus on how SRL is used in collaborative learning [63]. Efklides' "Metacognitive and Affective Model of Self-Regulated Learning" (MARSL) is one of the most recent SRL models, and is heavily influenced by the other models discussed here. It focuses on the interplay between metacognition, motivation, affect, and self-regulation [21]. For those looking for a more extensive discussion, Panadero et al. provide a detailed history and overview of the six models discussed in this section.

2.3.1 The 3 'waves' of Self-Regulated Learning Measurement. In their 2016 paper, Panadero et al. outline 3 major 'waves' of self-regulated learning was viewed by the research community and how those views impacted the measurement of this construct [64]. In the first wave, self-regulation was seen as static part of one's self. Measurements often focused on self-descriptive inventories like the MSLQ.

Over time, however, self regulation research has seen a shift in this mindset as critiques of self-reports arose. Chief among those concerns was that these self-reports more accurately measured student belief, rather than their real ability [75].

Panadero et al.'s second and third wave, self regulation was re-framed as a series of events to be witnessed, rather than in innate skill. This lead to a change in how the construct was measured. Rather than ask a student directly, researchers began to focus on watching for behavioral cues that could indicate a student was indeed self-regulating, like Think-Aloud studies.

While things like Think-Aloud studies did address this issue with student self-reports, research has moved into Panadero et al.'s 'third wave' of self regulated learning research. Here, the conception of self regulation as a series of events has not changed, but the focus has shifted to identifying ways to measure the construct unobtrusively. Researchers have started to explore what student interactions with digital tools can tell us about how a student self regulates their learning. Phillip Winne (co-creator of the Winne-Hadwin model) is a major advocate of studying self regulation using such data and has authored chapters in two recent editions of the "Handbook of Learning Analytics" [100] presenting how the COPES model can translate into an online environment. Indeed a

few studies within our corpus use data tracked from a class's learning management system (LMS) as an indicator for self regulated learning skills in action [11, 62].

2.4 Other Cognitive Control Reviews in Computer Science

As of this writing, no CS-specific models of cognitive control exist. The closest thing would be Loksa et al.'s model describing the self-regulatory steps that a student must engage with when problem solving [51]. These steps have been used to create Loksa and Ko's five types of self regulation that support problem solving when programming [50]. While these forms of self regulation did influence the skills identified, we seek to broaden our research beyond what students do when solving programming problems, and examine all skills a student would use when learning computer science.

Within CS, researchers have largely drawn from general education models to inform their understanding. Loksa et al.'s recent literature review provides an in-depth examination of how different works within CS education have effectively leveraged the theories of Zimmerman, Pintrich, and Flavell (among others) to better understand cognitive control when learning to program [52]. Loksa et al discuss cognitive control theories outside what has been seen in the Computer Science Education community with an eye towards which theories could be useful in computing education. This review also may be an excellent resource for those looking towards what new ideas can be brought into the field. In addition to discussing cognitive control theories within CS, the authors of this work discuss how cognitive control has been studied outside of the CER community with an eye towards which theories could be useful in computing education.

Prather et al.'s 2020 paper on cognitive control also serves as another in-depth exploration of the field. This review provides another exploration of how cognitive control is conceived of and measured both within the CER community and outside of it [75, 101].

A taxonomy of skills developed by Zimmerman and Martinez-Ponz has been used by two literature reviews to map the skills they observed to a general education model. A systematic review conducted by Silva et al., explored how interventions sought to improve self-regulation behaviors [87]. With an intervention addressing each phase seen in over half of their corpus, Zimmerman's three phases of *forethought*, *performance*, and *self-reflection* were all well represented. However, they noted that several strategies included in the Zimmerman and Martinez-Ponz taxonomy were not present in their review, including "environment structuring", "rehearsing and memorizing", and "reviewing texts". Silva et al. also observed that while self regulation was associated with success overall, efficacy was "not consistent across studies" [87]. Garcia et al.'s review of e-learning tools for Computer Science used the same taxonomy to identify areas where these tools were supporting self-regulation skills [28]. Ultimately, they noted "self-evaluation" and "goal setting and planning" were the most well-supported skills while "environmental structuring" and "seeking social assistance" were least supported. Additionally, while not part of the original taxonomy used for categorization, the authors noted *Emotional Regulation* was an emergent theme of e-learning tools that supported self regulation. While their work is similar, Silva et al. focused on studies detailing interventions targeting SRL, while Garcia et al. focused on software that supports SRL in some way.

In the present study, we expand our scope to research that examines self-regulation or metacognition in any context, rather than focusing on only interventions as previous authors do. We also do not start from the set of skills identified in the Zimmerman and Martinez-Ponz taxonomy as it was created from qualitative interviews with high school students in 1986. This is considerably different from the population of students we examine, which is modern college students learning to program. Modern students also have access to different resources than those available in 1986, most notably the internet.

3 Methods

To answer our research questions, we broadly surveyed the literature examining cognitive control skills (and their indicators) in post-secondary Computer Science education. We follow Plüddemann et al.'s methods for a Rapid Review [73], which adapts a systematic review approach to a more limited reviewer pool while still seeking to mitigate as much bias as possible. This means while every effort was made to follow the PRISMA 2020 standards for systematic reviews, one researcher was available to select studies, and extract data. In an attempt to limit potential biasing effects of the Rapid-Review process, another member of the research team verified a random sampling of papers after each filtering step. The authors also sought to establish a set of clear rules for filtering at each phase to minimize subjective judgments during study selection and data extraction. While resources were limited in terms of availability, the authors ensured that all qualitative coding was performed by multiple authors to help reduce bias at that critical point.

To answer **RQ1**, the research team read through each paper in the corpus and made note of passages where skills or behaviors were explicitly recognized as either self-regulation or metacognition. These passages were then qualitatively coded to indicate references to cognitive control skills, behavioral indicators, and discussions of difficulties. To answer **RQ2**, the research team also noted what (if any) relationship to academic success a given paper claimed for any skill or behavior.

Figure 1 provides an overview of how we selected studies for our corpus.

3.1 Eligibility Criteria

In order for a paper to be eligible for inclusion in our corpus, it had to have all of the following characteristics.

(1) The work must either be in English or have an English translation.
(2) Cognitive control had to be central to a research question or analysis of results.
(3) The work had have a clear research question or hypothesis that was examined through either experimental or quasi-experimental settings.
(4) The study's population must be focused on learning Computer Science at a post-secondary level. We believe cognitive control likely looks different for students experiencing the less-structured nature of post-secondary schooling than it does for students other environments.
(5) The work must identify specific skills (or behaviors) which represent Self-Regulated Learning.
(6) Skills (or behaviors) must be measured or observed in the paper itself, not just within cited works. In particular, this means that we did not consider literature review papers.

The numbers from this criteria list will be used as identifiers throughout the discussion of our methods to explain why papers were filtered at each stage.

3.2 Initial Search

To the best of our ability, we sought to generate a corpus broad enough to accurately capture the state of study on cognitive control in post-secondary CS education. Therefore, we wanted to ensure we were including all major publications of CS education research. To create a preliminary set of papers we created and refined a search string to use on the ACM Full Guide to Computing Literature online library. This database is described as the "most comprehensive bibliographic database focused exclusively on the field of computing" [27] and includes both journal and conference publications from the ACM and IEEE presses and other outside journals. A detailed list of all included publication sources can be seen at the ACM-DL website [27]. This database allowed our team to use one search query, rather than face the complication of replicating the query on multiple incompatible databases.

Our search query is shown in Figure 2. This query was adapted from those used in two other recent systematic literature reviews on cognitive control in computer science education [52, 75]. Using the ACM Digital Library's updated search feature, this string searched the full text of all studies by default - title, abstract, text, and citations

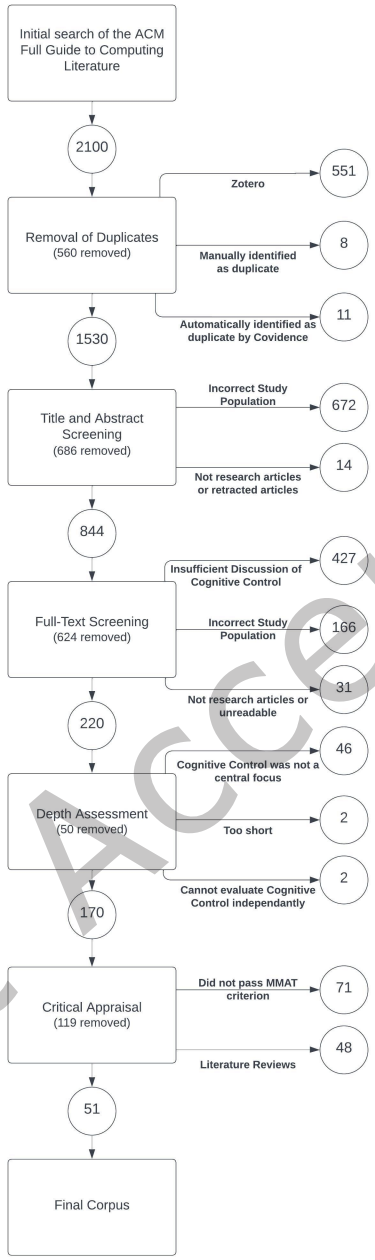

Fig. 1. Study Selection Diagram

("program" OR "programs" OR "programmers" OR "programming" or program*) AND ("metacognition" OR "metacognitive" OR "metacognitively" OR metacog* OR "self-regulation" OR "self-regulated" OR "self-regulatory" OR "self-regulating" OR self-reg*) AND ("computer science education") AND "learning"

Fig. 2. Updated search term used on the ACM Full Guide to Computing Literature

- for matches to these terms. All papers returned this way had to have some variant of the word "programming" and some variant of either "metacognition" or "self regulation". We limited our search to only include papers the ACM-DL categorized as research articles to exclude several kinds of publications that could not examine cognitive control in sufficient depth. These included proposals, posters, and published abstracts.

This search string includes both wildcard searching (such as "self-reg*") and explicit variants of the word. This is because the ACM digital library search tool does not support searching for variations of keywords automatically. The guide for using this search tool suggests using wildcards (like self-reg*) to search for variations of a word. However, after some experimentation, the research team found that only using explicit variants of a word (like "metacognition" vs "metacognitive") produced results not returned when only using wildcard searching. This effect also happened in reverse: using the wildcard self-reg* produced results not returned when searching for explicit variants of the word. Thus, both styles of search terms were included for "metacognition" and "self-regulation" in order to help identify all relevant papers.

We also found that the term "self regulation" was sometimes used outside of the study of education entirely. Studies where a computer would be regulating its own decisions (self regulation) rather than a governing body (government regulation) composed a significant percentage of this initial search. We found adding "learning" to the search query helped return works relating to learning programming.

Our search was limited to papers on or before December 31st 2023. There was no limitation on how old a publication could be.

3.3 Study Selection Process

Due to limited time, only one author was able select studies for this work. The potential implications, and the ways the authors have attempted to mitigate the potential bias introduced by this choice are discussed in Section 3.6.

3.3.1 Removing Duplicate Papers. The initial search yielded 2100 works. Citations were bulk downloaded in groups of 50 and imported into the citation manager Zotero. This software identified many papers that had at least one (sometimes two or three) duplicates. Using Zotero's de-duplication tool, we manually confirmed each potential duplicate from Zotero and removed 551 studies from our pool.

The remaining 1549 works were imported into the literature review tool Covidence. Before screening began, Covidence identified 11 additional papers that had more than one entry. After manually confirming these were duplicates, they were also removed.

3.3.2 Title and Abstract Screening. Our corpus at this phase consisted of 1530 works. In the next step, the reviewer examined only the titles and abstracts of each work to quickly remove clearly irrelevant papers.

The following criteria were used to exclude papers at this stage:

- The abstract explicitly notes that self regulation or metacognition are not a primary area of focus, violating eligibility criterion 2.

- The abstract explicitly notes that the students in question were not studying Computer Science in a post-secondary level class, violating eligibility criterion 4.
- The abstract indicates the work is something other than a research paper (i.e. a book review or blog post)

At this stage of screening, only obviously ineligible papers were excluded. In ambiguous cases, the paper was retained for the next stage. 672 works were excluded for studying populations outside of post-secondary education (such as K-12 or graduate students), or for setting their study within a non-CS course.

For our work, we wanted to focus on published works that sought to examine a research question in an experimental or quasi-experimental setting (as established in Criterion 3). At this stage, we discovered a handful of secondary sources like magazine articles, editorials, or blog posts. We opted to exclude these works as they were in violation of Criterion 3. Also excluded at this stage were summaries of working group meetings as they also did not explore hypotheses through experimentation. Eleven works were excluded as they did not fit into this definition.

Three additional papers were removed from the corpus because the publisher had retracted the work.

In total, 686 works were removed at this phase, 844 papers remained.

3.3.3 Full Text Screening. Our first phase of full text screening focused on ensuring that all papers fulfill eligibility Criterion 6. We wanted to filter out papers that make only a passing reference to cognitive control. We also wanted to ensure all works in our corpus addressed the experiences of students learning Computer Science at a post-secondary level. We first performed a vocabulary screening on the full text of each paper, and then read through the methods section to learn about the study population.

To conduct the vocabulary search, we opened each article and searched for the relevant vocabulary to determine how frequently self regulation and metacognition were discussed in the text of the work. To do this, we searched for the roots of our wildcard strings "self-reg" (with and without a hyphen) and "metacog" in each paper. Only instances of this vocabulary in the text of the work were counted. Instances of these words in the works cited, tables, keywords, or abstract were not counted towards a paper's vocabulary total. If the terms appeared at least twice, the paper was included. A paper would also be included if it made exactly one reference to metacognition and one to self regulation. Any work where either metacognition or self regulation were only used once in the text of the study were discarded for violating Criterion 6. This led to 427 disqualifications.

One member of the research team read through each remaining paper to determine whether the methods met the qualifications laid out in Section 3.1. The population needed to consist fully of post-secondary students, or the presented data needed to be stratified such that the post-secondary students could be evaluated. Additionally, the course these students took needed to be a post-secondary Computer Science course. Using this filtering process, 166 works were disqualified from our corpus.

To ensure our conclusions were generalizable, we also sought to ensure our corpus composed of studies that examined SRL from the perspective of experimental (or quasi-experimental) research. This imperative is discussed in further detail in Section 3.3.5. Some publications such as published abstracts, summaries of working group meetings, and case studies were also excluded at this stage as they did not employ an experimental or quasi-experimental approach. We opted to limit our scope in this way to ensure that the themes we saw works in our corpus were generalizable to other contexts. However, in excluding these types of works we do not wish to imply that such publications are unimportant: all offer valuable insight into cognitive control and all play an important role within our research community. Overall, we found 10 works to be in violation of Criterion 3 and removed them from our corpus

21 works were categorized as 'unreadable', either because no full text version could be found or because no English translation could be found, violating eligibility Criterion 1. In total 624 works were disqualified at this stage, leaving 220 papers in our corpus.

3.3.4 Evaluating Depth. We then confirmed that each remaining paper had a sufficient focus on cognitive control. This was necessary in order to satisfy eligibility Criterion 6. To accomplish this, we read every remaining paper to determine if cognitive control was a focus of the work. Eligible papers either measured or observed some cognitive-control skill through the course of their study. This means a paper had to discuss cognitive control in the methods, results, or discussion sections. If a paper wrote extensively of a cognitive control construct, but *only* did so as part of the background or introduction sections, it was removed from the corpus. This evaluation disqualified 46 papers from our corpus.

Following the work of Prather et al. [75], the research team chose to exclude papers that were less than two pages and papers that were works in progress as they could not examine cognitive control in sufficient depth. This disqualified two additional papers, leaving 170 in our corpus.

3.3.5 Critical Appraisal of Methods. The last step in creating our corpus was to conduct a critical appraisal of the methods. This was an integral step in the PRISMA review guidelines, to ensure each work in our corpus has a valid and trustworthy approach. In order to perform this assessment, we opted to use the Mixed Methods Appraisal Tool (MMAT) [74]. This tool was chosen as it offers support for methodologies that are common to education research like mixed method studies or quantitative studies on an un-randomized population. The 2018 version of the MMAT was used for this assessment.

This toolkit starts by evaluating all papers on two general standards. First, the MMAT asks: "Are there clear research questions?" Second, it asks: "Do the collected data allow to address the research questions?". Papers that qualified were then evaluated on five methodology-specific standards. These centered on ensuring that data were collected, reported, and synthesized according to the standards of the methods in question. Studies that were mixed-methods in nature were evaluated using the standards for all relevant methodologies, and on a separate mixed-method specific criteria list. 71 works were removed for not fulfilling the MMAT toolkit standards.

46 of those 71 (64.78%) were removed for not having a research question or other stated goal for the work. Over one third (71 removed from the remaining 170) of the remaining pool of papers was removed due to the MMAT. Many of these papers removed at both of these phases were not necessarily weak, but had other concerns beyond attempting to answer a question through experimentation. For example, many of these works were presenting novel software to the community, rather than exploring existing phenomena.

48 literature reviews were also disqualified as they violate eligibility Criterion 3. While such works could have been excluded in the title and abstract screening based on this criterion, the authors were initially hesitant to exclude literature reviews based on this criterion as they still do answer a research question using a standardized approach. Ultimately, however, after reviewing the wording of Criterion 3, it became clear that literature reviews did not utilize an experimental or quasi-experimental method and were closer to secondary sources.

Still, these other reviews are pertinent to this work. A number of these literature reviews are highlighted in Section 2.

After completing these steps, 119 papers were removed and we were left with 51 research studies. The full list of included works can be seen in Appendix A.

3.4 Data Extraction

Once the final corpus of works was created, the next step was to read over every study and extracted of the relevant passages. Again, this process was conducted by only one member of our research team due to limited time availability. Section 3.6 provides more discussion as to the potential bias this introduces.

We used the following criteria to extract data that was relevant to answering our research questions:

To be relevant to RQ1, a passage needed to make it clear what they considered to be a cognitive control skill, or be related to an already clearly defined skill. For example, "Learners may self-regulate by

being aware of limitations in their knowledge of related problems" is a passage from Loksa and Ko's study [50]. Here, awareness of limitations in knowledge is explicitly mentioned as an aspect of self regulation.

After this key statement, any sentence going into further detail describing this form of awareness was also extracted. For example, Herman et al. defined metacognition in their codebook as : "subject shows evidence of monitoring what they are doing and checking their work" [38]. Here, 'monitoring work' was explicitly called out as part of the definition of metacognition. Therefore, the research team member extracting data also extracted passages that further defined what 'monitoring work' meant within the context of this paper. This lead to the extraction of the following passage: "During all of the interviews on Boolean word problems only four instances emerged where subjects monitored the accuracy of their work or strategies. In particular, after subjects had derived boolean expressions they rarely returned to the original English expressions to check the accuracy of their results" [38].

Similarly, authors often created tables listing a number of self-regulation skills, like those seen in Cristea et al. [11], Prather et al. [76], or Falkner et al. [22]. All skills from such a list were extracted, as were any sentences from the text of the paper that further described the listed skills.

Our focus is on cognitive control skills studied in CS education research. We only extracted passages when authors clearly were drawing a connection between cognitive control and a skill. This criterion was chosen because it helped reduce subjective judgments when extracting passages. Inferring whether a vague passage was indeed talking about metacognition or self regulation had the potential to bias this set of passages to the author's notions of what did or did not count. To keep this potential introduction of bias to a minimum, the author performing data extraction only considered skills that were explicitly connected to self regulation or metacognition by the text of the paper.

A passage could be of any length, though ultimately none were longer than a paragraph. Four authors were involved in qualitatively coding these passages. The process of qualitative coding is described in more detail in Section 3.5.

In total 216 passages were extracted for qualitative coding to answer RQ1.

To answer RQ2, one member of the research team read over each paper again and made a determination if the **paper's outcomes claimed to connect some aspect of cognitive control to academic success**. In order to reduce bias, this was done by noting at least three passages from each paper. To clarify this process we will first detail the general process and then provide an example of what passages were extracted from Denny et al.'s 2019 study.

First, at least one passage from the original collection of 216 passages had to note that a particular concept was an aspect of cognitive control. The first passage extracted from Denny et al. was: "...students who misinterpret a problem statement and don't have the metacognitive awareness to identify when their mental model deviates from that statement may struggle to make progress towards a working solution" [13]. This was used as evidence that Denny et al. saw 'misinterpreting a problem statement' as a behavioral indicator for missing metacognitive awareness.

Second, the paper must contain at least one passage (not necessarily in our collection of 216 passages) demonstrating that the authors were connecting cognitive control constructs to success. We extracted the following passage from Denny et al. at this step: "In this research, we explore the effect of this experiment prompt – which provides metacognitive scaffolding around interpreting the problem statement – on student performance on the corresponding programming task" [13]. This sentence indicated that the authors were relating metacognitive awareness (via scaffolding on experimental problem prompts) to success on those tasks.

Third, the paper must contain at least one passage (not necessarily in our collection of 216 passages) describing the relationship observed. This allowed us to note when the skills and indicators from RQ1 were connected with success, and gauge the significance of that connection. Two passages were extracted from Denny et al. to fulfill our third step for this paper. The first passage was: "...the results of our study did not confirm any

significant difference in the overall categories of completion time, completion rate, or number of attempts before achieving a correct solution when comparing control and experimental groups" [13]. The second passage was: "students in the experimental group encountered fewer errors on average resulting from incorrect mental models compared with the control group, and this difference has strong statistical significance," [13]. Here, Denny et al. succinctly summarize their findings indicating which measures found a statistically significant difference between experiment and control.

No study was discounted if the examined skill did not have a measurable effect on success. The only requirement was that a work clearly connected at least one cognitive control skill to academic success, not that the data supported that connection.

Qualitative coding was not performed for passages extracted to answer RQ2. Rather, these passages were used to write a short summary of each finding reported by a paper. This helped the research team better understand how a skill related to academic success in each paper so overall trends could be presented here. A spreadsheet containing all passages extracted to answer our research questions will be available. This, and other supplementary documentation is discussed in more depth in Section 3.6.

3.4.1 Extraction From MSLQ. As mentioned in Section 2.3, nine papers (17%) of the final corpus sought to measure cognitive control through Pintrich's widely used MSLQ inventory [8, 9, 18, 33, 45, 46, 49, 76, 90].

In this inventory, students rate how true 81 statements are to their lives and learning habits. Not all of these questions were extracted and used for answering RQ1, however, as several subscales focus on measuring related constructs like motivation or critical thinking, rather than self regulation or metacognition. Following our general policy for extracting data for RQ1, we wanted to capture what the authors of each study thought were cognitive control skills. In some cases, the authors were very clear about what MSLQ subscales they used to study self regulation. Duvall et al. noted that they used both the Metacognitive Self-Regulation (MSR) subscale and the Effort Regulation subscale to measure self-regulation skill [18]. Leppänen et al. noted they used three subscales (MSR; effort regulation; and time and study environment management) to measure self-regulation skills [45]. When studies made no mention of which subscales they used to measure self regulation, it was assumed the MSR subscale was used as it most closely matched our working definition. The same assumption was given to Han and Ellis' 2023 study which notes that it uses 2 subscales: "positive self-regulated learning strategies (4 items) and negative self-regulated learning strategies (3 items)" [33] from the MSLQ's 1991 version of the test. To our knowledge the MSLQ-1991 possesses no such subscales. In this case, it seemed most likely they were using a subset of 7 questions from the MSR subscale but no questions were specified in the work.

Multiple versions of the MSLQ were used by papers in the corpus. Most used the MSLQ from 1991 [69], which we will refer to as MSLQ-1991. This makes sense as it is the most well validated and most commonly used outside of CS education work as well [71, 83].

Castellanos et al. used MSLQ-Columbia [81], which is a Spanish translation of MSLQ-1991. It also re-organizes the MSR subscale into three smaller metacognition-focused subscales, using 9 out of 12 questions from the MSLQ-1991 MSR subscale. These smaller subscales are: Planning, Monitoring, and Study Method Regulation [81]. Due to resource limitations, the original Spanish text could not be coded for this work. Instead, Ramírez-Echeverry et al.'s work [81] discussing the translation process was referenced. No lexical changes between the MSLQ questions and MSLQ-Columbia questions were reported by Ramírez-Echeverry et al. Therefore, coding team for this review coded the MSLQ-1991 in English and noted where a passage was translated for use in the MSLQ-Columbia.

The slightly older MSLQ-1990 was used in two studies. This is similar to MSLQ-1991, but all questions are slightly re-phrased. Thus, all questions from the sub-scales detailed in Table 1 were extracted as unique passages and coded independently to help answer RQ1.

Citation	MSLQ Version	Subscales Extracted
[18]	MSLQ -1991	Metacognitive Self-Regulation Effort Regulation
[46]	MSLQ -1991	Metacognitive Self-Regulation
[9]	MSLQ -Columbia	Planning Monitoring Study Method Regulation
[76]	MSLQ -1991	Metacognitive Self-Regulation (3 questions)
[45]	MSLQ -1991	Metacognitive Self-Regulation Effort Regulation Time and Environment Managment
[8]	MSLQ-1990	Metacognitive Self-Regulation
[49]	MSLQ -1991	Metacognitive Self-Regulation
[90]	MSLQ-1990	Metacognitive Self-Regulation
[33]	MSLQ -1991	Metacognitive Self-Regulation

Table 1. Version and subscales extracted from each paper that used the MSLQ

3.5 Data Synthesis

After the data extraction phase was complete, the research team performed qualitative coding on all 216 passages that identified a cognitive control skill or behavior. Each passage was given one of 33 codes describing cognitive control skills, behavioral indicators of those skills, or situations where students had difficulty with using those skills. Our finalized code book can be seen in Table 3.

General Process: Passages were assigned to different members of the coding team such that at least two coders reviewed each passage. The process of assigning passages will be described after going over the process the team used to assign codes.

To start the process, a team member would review the collection of passages assigned to them, each within the context of its originating paper. The team member would then write short summaries for each passage. In qualitative analysis this process is called 'memoization', and it helps a coder gain familiarity and identify trends with all passages before starting to assign codes.

Next, each team member would independently assign a code to each passage. The team would then convene to discuss how each team member had coded each passage. Team members had the opportunity to change what code they had assigned if they felt another team member's code was a better fit. If the two coders could not reach an agreement, a third member was asked to review the code in context and assign their own code. In most cases, the tie-breaker reviewer would agree with one of the codes that the two original reviewers had suggested. In rare cases, the tie-breaker reviewer would chose a third code that they felt better fit the passage. The three coders would then meet, return to the paper, and reread the passage in context together. They would then come to a determination that all three could agree on.

Work Assignment: There were four members of the coding team overall, but they never worked together all at once.

Three reviewers coded the first 99 passages. The team of three was initially assigned 20 passages and followed the process described above as a group in a calibration phase. This exercise helped all team members gain a sense for the coding process, calibrate understanding of what the preliminary codes meant, and identify where new codes might be needed.

Following the calibration phase, the next 79 passages were memoized, coded, and discussed. A code was assigned to a passage if at least two out of the three members of the coding team assigned the same code to it.

After all codes had been assigned to these 99 passages, the team structure shifted. One member of the original coding team was unable to continue working, while another had reduced availability. To address these constraints, a fourth team member was added and assignments for the remaining 120 passages were adjusted. All passages were assigned such that they received an initial code from at least two team members. As with the first round of coding, the three team members initially worked together in a calibration phase to assign codes to 24 passages to calibrate their understanding. Following that, the same process of memoization, coding, and discussion was used to complete the process.

Agreement: Table 2 describes the percent agreement between all pairs of coders that worked together. The team member labelled 'R3' is the one who could not participate in coding the second half of the passages. 'R4' represents the new team member who took their place. Since R3 and R4 never coded a passage together, they have no agreement to report.

Percent Agreement Before Discussion

	R1	R2
R1		
R2	62.33%	
R3	52.17%	30.43%
R4	81.82%	83.33%

Percent Agreement After Discussion

	R1	R2
R1		
R2	72.08%	
R3	66.30%	41.30%
R4	80.56%	76.67%

Table 2. Percent agreement between all pairs of coders

As there were several reviewers, all with varying levels of overlap with each other, we opted to use Krippendorff's Alpha [42] for nominal data to calculate inter-rater reliability. Marzi et al.'s K-Alpha calculator was used to compute this score [60]. The K-Alpha for our code book was 0.897 which, according to Marzi et al., indicates a satisfactory level of agreement and a reliable rating.

Final Review: After each passage had been assigned a code, the coding team met once more to review and discuss if all passages had been assigned the best possible code. The list of codes and corresponding assigned passages went through considerable refactoring at this stage as trends in skills and behavioral indicators emerged.

Passages were then re-assigned to these refactored codes based on where they fit best. For example, "Forming the wrong conceptual model about the right problem" was a passage from Prather et al. [78]. Initially this was coded as *Task Analysis*. However, in this refactoring phase, it became clear there was a trend of passages more specifically discussing situations where students struggled to effectively analyze a task. Therefore, the coding team created the sub-category: '*Task Analysis* - difficulties' and re-assigned some passages to this new sub-category.

Skills were also grouped together into overall skillsets that loosely related to when in the work process that students might use them. Section 4 describes the identified skillsets in more detail.

After this refactoring, the remaining three members of the coding team reviewed all passages, discussed the new assignments, and reached 100% agreement on the assignment of codes to all passages.

A list of all codes and definitions can be seen in 3.

Use of Theory-Generated Codes: Before starting to code, all members of the coding team were given an overview on the basics of cognitive control by reading through three works to get an overview of the important ideas. To do this, all team members read over Loksa and Ko [50], Prather et al. [75], and Garcia et al. [28].

Initially, the codes for these passages were created inductively, meaning the team read through passages and created their own codes and definitions to suit the text. However, it became clear even after training that many of

the codes created at this stage were ill-defined and did not have a good mapping to existing cognitive control theory.

In order to effectively answer RQ1, we wanted to ensure that the skills we identified had a solid grounding in existing theory. We therefore introduced the list of skills from Domino et al. [17] as a list of theory-generated codes to get started. In this context, a *theory-generated code* is one derived from existing literature, and is distinct from an *in-vivo* code, which is created by reviewers while coding a text [15, 58]. The skills from Domino et al. were selected as the set of terms to start with as they define a set of cognitive control skills that each have a firm grounding in both educational theory and existing study within CS education research.

Theory-generated codes act as a way to provide a consistent and reasonable starting point for qualitative coding, but are never meant to take the place of codes derived from the text. Members of the coding team were strongly encouraged to see these initial codes as optional. They were directed to create new codes in situations where none of the theory-generated code seemed appropriate. They also had the option to mark any passage as "unsure" if they wanted additional information about existing theory before assigning a code.

The coding team took these directions to heart and did not simply use the list of skills from Domino et al. [17] to deductively code data. They added many new codes and removed unhelpful theory-generated codes. In the final review phase discussed in Section 3.5, the coding team also grouped these skills differently from Domino et al.. While we believe the use of these theory-generated codes as a starting point was an appropriate choice for answering RQ1, this approach had an undeniable impact on our answer to this question. A more in-depth discussion on the biasing effects of this decision and the ways we have further sought to address bias is addressed next.

3.6 Areas of Potential Bias and Artifacts of this Work

As this is a Rapid Review, which can be more biased due to the more limited resources available, we seek to be as transparent as possible about areas of potential bias within this work. To that end, this section describes places where bias was potentially introduced into our work.

One significant limitation in our approach was that only one author was available to perform study selection and data extraction. This work was conducted without grant funding, meaning time to devote to this study was limited throughout much of this work's timeline. It was only when passages were ready for synthesis were other authors available to assist in this work. While we do highlight areas where subjectivity clearly came into our decision making process, there are almost certainly places where this author unknowingly introduced bias into our work. This study's limited resources for conducting aspects of this review were a known factor before the review began. It was for this reason we opted for a rapid review methodology rather than a full systematic review.

The best strategy we see to help mitigate this issue is transparency. In addition to thoroughly describing our approach, we also provide publicly available artifacts that document our process. In doing this, we hope that future researchers will be able to replicate and refine our approach to better combat bias within results. All of the additional artifacts described in this section can be found at https://github.com/ProfDomino/-TOCE-Literature-Review-Supplementary-Documentation#.

Study Selection Bias: Bias was likely introduced in searching and selecting studies for this corpus.

While the ACM Full Guide to Computing Literature online library is extensive in breadth, it does focus on *computing* literature. Thus, our resulting corpus focuses on only the works published at some computing-focused venue. This was preferable as it helped us more effectively identify research that studied the experience of post-secondary CS students. However, it is possible relevant papers were missed in our initial search because they were published in related domains outside of computing, like education research.

Throughout the process of study selection, we emphasized the use of certain vocabulary words. Our search string was formed to look for variants of 'self regulation' and 'metacognition', and our full text screening

disqualified papers that did not use these terms enough. Hypothetically, a highly valuable study on help-seeking that made no mention of cognitive control, would not be returned in our initial search. Thus, the results presented here reflect the state of cognitive control research, not the state of the art on any individual skill.

Subjectivity was likely introduced when conducting the critical appraisal of bias. The first question within the MMAT toolkit is: *Are there clear research questions?*. This may seem straightforward, but there were cases where only examining a paper with research questions would have been overly limiting. For example, Herman et al. [38] state: "The goal of this paper is to understand the ways in which students fail to translate accurately from English to Boolean expressions". Even though it is not phrased as a question, the research team read this as a clear statement of research aims and the approach taken supports this goal. Such determinations inherently involve some level of subjective judgement.

Other questions from the MMAT like *"Do the collected data allow to address the research questions?"* or *"Are findings adequately derived from the data?"* introduce subjectivity as well. A random sampling of these decisions was validated by a second researcher (with 100% agreement), but it is possible that the paper selection process introduced some level of bias within our work. With this in mind, we have created a document detailing every paper in our corpus (after the de-duplication stage) as well as details about when and why it was excluded.

Overall, we feel we have constructed the best possible corpus given our constraints. Our results demonstrate that a diverse set of skills are being examined in cognitive control research. If a study was indeed incorrectly included (or excluded), it might impact our observations about which skills are seeing a lot of focused study, but we feel it is unlikely that such a change would have much effect on which skills were observed.

Data Extraction Bias: When extracting passages for RQ1, we attempted to reduce subjective decision making while extracting data by only identifying skills clearly called as self regulation or metacognition. While we chose to do this to reduce subjective decision making, there was a limit to how effectively we can eliminate judgment calls from this process. As described in Section 3.4, we sought to identify passages where authors explicitly noted a skill was an aspect of metacognition or self regulation. Still, cases where it was unclear what skills a study was referring to did come up.

For example, Peteranetz et al. [68] outline four aspects of "student self-regulation and strategic engagement" in their work. Those aspects are: "general metacognitive self-regulation", "Knowledge Building", "dysfunctional self-regulatory strategies", and "behavioral engagement with the class" [68]. While the term self regulation is directly in two of these elements, it is unclear if the authors see *Knowledge Building* or behavioral engagement with the class as a component of self regulation, a component of strategic engagement, or a component of both ideas. Ultimately, it seemed safest to cast the widest possible net and extract passages relating to all four concepts. While we feel that this was most appropriate for answering our research questions, it is possible Peteranetz et al. defined self regulation more narrowly. While such judgements were not common they were a component of this review. Generally speaking we did our best to read closely and extract everything that seemed like it could be related. However, it is possible we have misunderstood the author's understanding of cognitive control and included constructs that are related but external to their definitions of self regulation or metacognition.

To be as transparent as possible about this inevitable form of bias, a spreadsheet detailing every extracted passage to answer either research question is available with our online documentation.

Bias within Data Synthesis: Another potential area of bias was the decision to provide the coding team with the list of skills from Domino et al. [17] as a set of theory-based codes. Had the coding team worked solely from the extracted passages, it is likely the resulting list of skills would be different. However, we believe it is also likely such a list would carry a much weaker basis in existing theories of cognitive control, making it a much weaker list overall. Thus, while our methods are sufficient to answer our research questions, it did mean our coding team was likely biased in creating this particular list of skills.

To be as transparent about how our coding team utilized the list of skills from Domino et al., we have created a document that serves as a change log. This document details how our theory-based codes developed into our final set of codes. The diagram of the final set of codes presented in this research can be seen in Figure 3. Diagrams of the initial theory-based skills from Domino et al. [17] and a copy of Figure 3 can be found in Appendix B for easy comparison.

Language Bias. As the author screening papers and extracting data only speaks English, only papers originally written in English or with a translation to English could be included in this work. Ultimately, only a handful of works were excluded for this reason, but a survey of a more diverse set of studies in a wider range of languages could be a valuable area for future work.

4 Identified Skills

This section provides a discussion of the 11 cognitive control skills and 10 behavioral indicators of a skill in action that were identified from the study corpus. Specifically, these skills and behaviors where identified from analysis of the 216 passages extracted from our corpus and coded as described in Section 3.5. Recall that each was given one of 33 codes describing cognitive control skills, behavioral indicators of those skills, or situations where students had difficulty with using those skills.

As described in Figure 5, the identified skills are mostly grouped into four overall skillsets. To keep things clear, skillsets will be in bold while skills will be italicized.

Table 3 presents our working definition for the skills along with the behaviors used to indicate a given skill. Table 4 shows references to all papers that discuss a given skill, behavior, or difficulty.

For ease of presentation, our skills and indicators are organized into a hierarchy, with related skills collected into skillsets. Our skillsets match the three common phases of self-regulation models identified by Puustinen et al. [80]. These phases are the preparatory phase, the performance phase, and the self-control phase. All three major theories of self regulation discussed in Section 2.3 have analogues Puustinen et al.'s phases.

In the preparatory phase, students set intentions and strategize. We represent this with our **Planning** skillset where *Task Analysis*, *Scheduling*, and *Decomposing* are discussed. In the performance phase, students self-observe and control their behavior, which we divided into two distinct skillsets. Self-Observation is reflected in our **Monitoring** skillset where students maintain an awareness about their correctness, completeness, and emotional state. Self-Control is reflected in our **Adaptation** skillset where students adjust their approach based on their observations by seeking help or regulating their environment to reduce distractions. In the appraisal phase, students react and reflect. This is encompassed in our **Reflection** skillset where students think back on the work they have done and make larger-scale plans for the future.

One skill and one indicator did not fit into any of these four skillsets or three phases, as they could occur at a number of different points in the process of preparation, performance, and appraisal. *Knowledge Building* could occur while a student is preparing to work, while working, or after finishing. Similarly, procrastination is marked as a 'generalized indicator' because it is a behavior that might result from difficulties associated with many skills. These two cases will be discussed separately from the other skills and indicators on this list.

In Table 4, references never represent a super set of other codes. For example, the two papers listed in the second row, *Task Analysis*, discussed this skill. It so happens that both of those papers also discuss difficulties with *Task Analysis*, so they are also listed on the third row. Of the two, only one paper, Falkner et al. [22], also discusses reinterpreting materials as a part of *Task Analysis*. Some papers contain passages that were coded as a skillset, rather than any specific skill or indicator. These represent situations where, even within the context of the paper, a skill from our list was never clearly defined. For example, Peteranetz et al. list out their relevant components of self-regulated metacognitive strategy, but do not elaborate on what they mean by 'planning' [68]. It could refer to a student determining *how* they intend to approach a problem, but it could also refer to *when* a

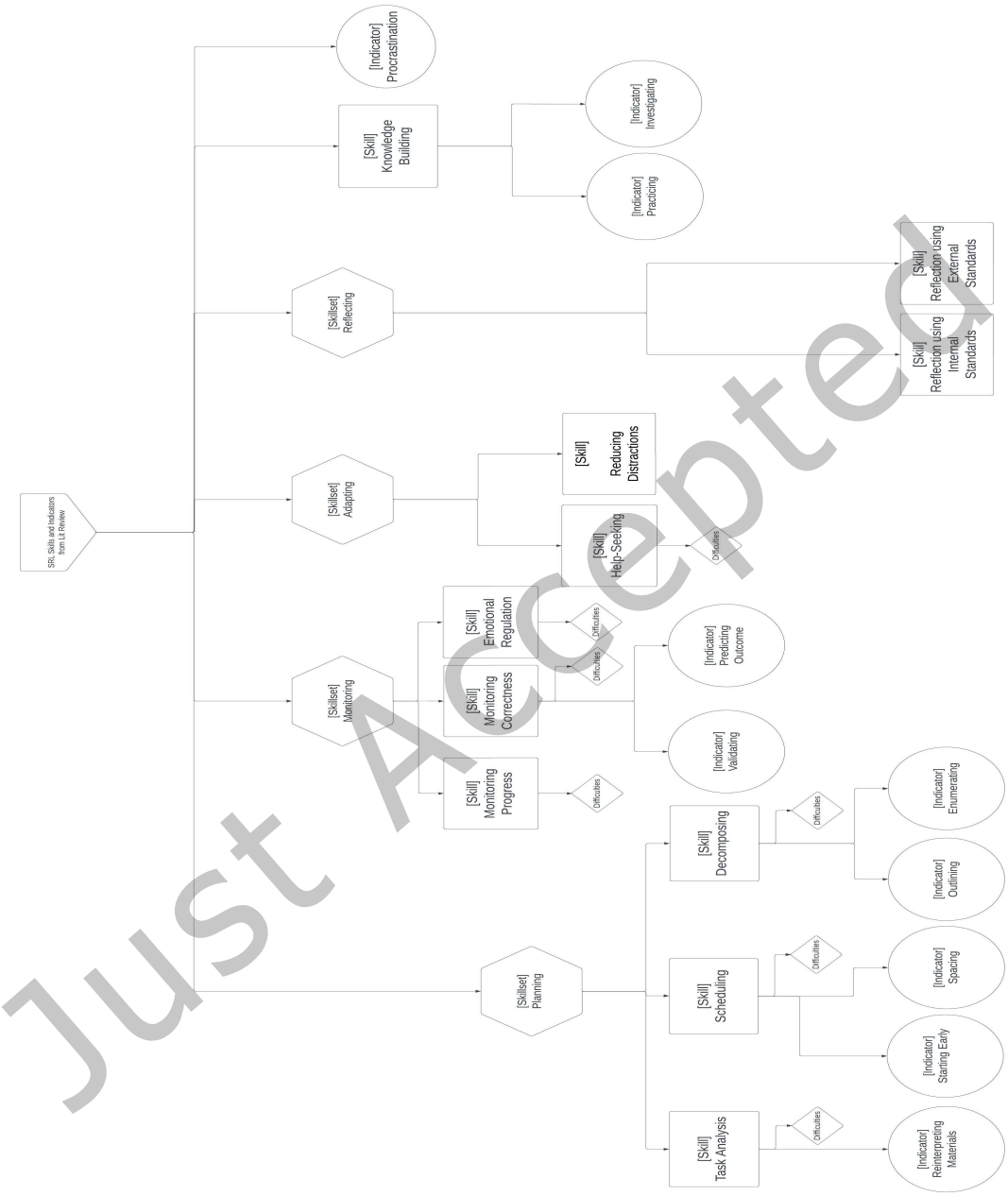

Fig. 3. Hierarchical description of all identified skills and their skillsets

Code	Definition
Planning (skillset)	General discussion of 'strategizing', typically as a first step to completing a task
Task Analysis (skill)	Forming an accurate conceptual model of the task at hand
Task Analysis - difficulties	Cases of students who formed an inaccurate conceptual model and their subsequent difficulties completing tasks
Task Analysis - Reinterpreting Materials (indicator)	Students understanding a problem by transforming or annotating assignment text
Scheduling (skill)	Students intentionally allotting blocks of time in the future to complete the task
Scheduling - difficulties	Cases of students not setting aside enough time to complete work before a deadline and their subsequent difficulties completing tasks
Scheduling - Starting Early (indicator)	Students scheduling by allocating time blocks to work that are well before an assignment is due
Scheduling - Spacing (indicator)	Students scheduling by allocating many smaller time blocks throughout the course of an assignment (rather than allocating one large block for all work)
Decomposing (skill)	Students taking an abstract task and breaking it into smaller more concrete sub-tasks as a way to construct an overall strategy or algorithm
Decomposing - difficulties	Cases of students who did not break a task into smaller components and their subsequent difficulties completing tasks
Decomposing - Outlining (indicator)	Students decomposing a problem by creating method skeletons before implementing
Decomposing - Enumerating (indicator)	Students decomposing a problem by creating a list of sub-tasks needed to complete the larger goal
Monitoring (skillset)	General discussion of 'maintaining an awareness of self
Monitoring Correctness (skill)	Students maintaining an awareness that the work they are doing is correct
Monitoring Correctness - difficulties	Cases of students who were unaware that the work they were doing was incorrect and their subsequent difficulties completing tasks
Monitoring Correctness - Predicting Confidence (indicator)	Observing a student's awareness of their correctness by having them self-report their confidence in their answers
Monitoring Correctness - Validating (indicator)	Students monitoring their correctness by taking steps to prove to themselves that the answer they came up with is correct
Monitoring Progress (skill)	Student maintaining an awareness of how close they are to completing their task
Monitoring Progress - difficulties	Cases of students who were unaware of how close they were to finishing and their subsequent difficulties completing tasks
Emotional Regulation (skill)	Students maintaining an awareness of their current emotional state and the potential impact those emotions could have on their work
Emotional Regulation - difficulties	Cases where students were students were not mindful of their emotions and their subsequent difficulties completing tasks
Adaptation (skillset)	General discussion of students making a strategic change to their work
Help-Seeking (skill)	Students seeking outside assistance after recognizing a need for that help. Involves assessing what questions to ask, what resources to seek help from, and when to reach out for help
Help Seeking - difficulties	Cases where students did not ask for help and their subsequent difficulties completing tasks
Reducing Distractions (skill)	Students making an effort to adjust their study environment so they do not become distracted
Reflection (skillset)	General discussion of students considering their approach to completed tasks (or subtasks) as it relates to work they will do in the future
Reflection - Reflection using External Standards (skill)	Students assessing their work in the context of course expectations
Reflection - Reflection using internal standards (skill)	Students assessing their work in the context of their expectations of themselves
Knowledge Building (skill)	Students engaging in*Knowledge Building* in ways that are not for credit within a course
Knowledge Building - Practicing (indicator)	Students building knowledge by working on ungraded practice exercises
Knowledge Building - Investigating (indicator)	Students building knowledge by seeking out answers to questions not directly related to the course
Procrastination (general indicator)	Observations of students who appear to intentionally delay working in spite of the negative consequences of that delay.

Table 3. All Codes and Definitions

student intends to work. All we can conclude is that the authors of this work identified some planning processes as an important component of cognitive control. The best fit for this passage was, then, the planning skillset rather than a specific skill.

Figure 3 shows a diagram of all identified skills, indicators, and difficulties grouped by skillset. It can also be found in Appendix B.

MSLQ. The Motivated Strategies for Learning Questionnaire is an 81-question inventory divided among 15 sub-scales. Since only some of these subscales were used to measure cognitive control, only the questions from relevant subscales were extracted for analysis in this study. A single extracted subscale from the MSLQ could cover a wide variety of the skills listed in Table 3. This makes sense as the MSLQ was designed to capture a holistic view of a student's ability to self-regulate. However, it does make reporting the skills that the MSLQ captures somewhat difficult as many papers use different versions of the inventory, different subscales, or even

Code	Code Count	Referenced In
Planning (skillset)	4	[9, 22, 68, 88]
Task Analysis (skill)	2	[22, 29, 45, 62, 79]
Task Analysis - difficulties	2	[22, 45]
Task Analysis - Reinterpreting Materials (indicator)	6	[2, 13, 22, 78]
Scheduling (skill)	10	[1, 9, 22, 55, 65, 67, 76]
Scheduling - difficulties	2	[2, 22, 45]
Scheduling - Starting Early (indicator)	6	[2, 11, 14, 41, 44, 104]
Scheduling - Spacing (indicator)	4	[2, 10, 41, 44]
Decomposing (skill)	15	[9, 18, 22, 46, 50, 55, 68, 76, 85] [8, 11, 24, 33, 45, 49, 90, 105]
Decomposing - difficulties	1	[2]
Decomposing - Outlining (indicator)	7	[22, 82, 94]
Decomposing - Enumerating (indicator)	3	[9, 18, 22, 45, 46, 49, 65]
Monitoring (skillset)	3	[38, 68, 91, 105]
Monitoring Correctness (skill)	21	[9, 18, 22, 38, 46, 50, 65, 86, 96] [8, 12, 33, 43, 45, 49, 77, 90]
Monitoring Correctness - difficulties	6	[2, 38, 47, 78]
Monitoring Correctness - Predicting Confidence (indicator)	5	[12, 43, 86, 91, 96]
Monitoring Correctness - Validating (indicator)	14	[9, 18, 22, 45, 46, 50, 55, 65, 82, 94] [8, 49, 90]
Monitoring Progress (skill)	3	[24, 45, 50, 62, 67, 68, 94]
Monitoring Progress - difficulties	2	[78]
Emotional Regulation (skill)	5	[18, 22, 33, 45, 46, 49, 76]
Emotional Regulation - difficulties	5	[8, 9, 76, 90]
Adaptation (skillset)	4	[9, 18, 22, 45, 45, 46, 49, 91]
Help Seeking (skill)	11	[2, 11, 16, 22, 29, 55, 68, 79, 82, 89]
Reducing Distractions (skill)	4	[9, 22, 45, 76]
Reflection (skillset)	3	[7, 33, 37, 55]
Reflection - Reflection using External Standards (skill)	3	[11, 18, 24, 45, 46, 49, 92]
Reflection - Reflection using Internal standards (skill)	2	[2, 9, 18, 40, 45, 46, 49, 103]
Knowledge Building (skill)	1	[68]
Knowledge Building - Practicing (indicator)	4	[8, 22, 55, 76, 90]
Knowledge Building - Investigating (indicator)	3	[22]
Procrastination (general indicator)	11	[2, 19, 22, 67, 85, 102, 104]

Table 4. Code Counts and References

subsets of questions within a subscale. Table 5 describes the skills that are captured in various forms of the MSLQ and references the works within this corpus that use this inventory directly.

We next discuss the various skills in the order that they are presented in Table 4.

Skill	Metacognitive Self-Regulation (MSLQ-1990)	Metacognitive Self-Regulation (MSLQ-1991)	Effort Regulation (MSLQ-1991)	Time and Environment Management (MSLQ-1991)	Metacognitive Planning (MSLQ-Columbia)	Metacognitive Monitoring (MSLQ-Columbia)	Metacognitive Study Method (MSLQ-Columbia)
Task Analysis							
Scheduling				[45]			
Decomposing	[8, 90]	[18, 33, 45, 46, 49]			[9]		
Monitoring Correctness	[8, 90]	[18, 33, 45, 46, 49, 76]				[9]	
Monitoring Progress				[45]			
Emotional Regulation	[8, 90]	[18, 33, 45, 46, 49, 76]	[18, 45]			[9]	
Help-Seeking							
Reducing Distractions				[45]			
Reflection		[18, 33, 45, 46, 49, 76]		[45]			[9]
Knowledge Building	[8, 90]						

Table 5. Skills Captured Within Different Versions of the MSLQ

4.1 Planning Skills

The corpus contains many skills that fall under the umbrella of **Planning**. The term was used to describe how a student understands a problem, strategizes their approach, and allocates time. The skills in Table 4 associated with this skillset reflect those three components. These skills all loosely fit within the *preparatory* phase of Puustinen et al.'s taxonomy [80]. Though a student may call upon them while working, they are most likely used before beginning to work in order to set intentions and strategize.

4.1.1 Task Analysis. *Task Analysis* is often discussed within this corpus through the lens of starting too quickly, though the skill does extend beyond this one behavior pattern. Novice programmers commonly demonstrate an eagerness to start writing code before they've fully thought things through. Denny et al. note: "...students who misinterpret a problem statement and don't have the metacognitive awareness to identify when their mental model deviates from that statement may struggle to make progress towards a working solution" [13]. Ultimately, *Task Analysis* concerns the critical practice of taking time to form a proper mental model of the task. This prevents misunderstanding requirements, which can ultimately cause students to get stuck.

While a correct conceptual model is the ultimate goal of *Task Analysis*, a student cannot actually *know* if their model is indeed correct. Therefore, correct knowledge of requirements is not an aspect of cognitive control by itself. What is critical is that a student thoughtfully and intentionally reads and processes instructions. If they do this and do not ultimately come up with a perfect understanding of their task, they have still engaged in *Task Analysis*. For example if a student thoughtfully and methodically reads over instructions, but misunderstands how a feature of their software should work, they still applied their skill at *Task Analysis* to the problem.

Within general education research, *Task Definition* is a component of the Winne-Hadwin model [97] and Zimmerman's Cyclical Phase Model [106]. Greene et al. highlight a trend of implicit assumption among instructors that problems have been written in a way that students will understand [31]. Malmberg et al. noted "empirical research has shown that learners do not always completely understand what the teacher or instructor expects for task accomplishment." [57]. Both authors highlight how, in order to succeed, students need to be able to identify metacognitive cues from a problem statement that help them understand what they are being asked to do. They also need to be able to synthesize such cues with their own perception of the task.

Seeing a student reinterpret materials is one indicator that has been used to identify CS students taking time to form that good conceptual model. For example, seeing a student review [55, 65] or annotate [76] a problem statement would be indicative of *Task Analysis* in action.

Researchers have observed during Think-Aloud studies that CS students struggle with *Task Analysis*. The authors of these works will frequently comment on times when students verbalized incorrect conceptual models [2, 76–78]. Prather et al. found formation of a correct conceptual model was the "most glaring inconsistency" between students who successfully completed their Think-Aloud assignment and those who did not [78]. In their code book for their Think-Aloud experiment, Loksa and Ko identified two types of "Comprehension Monitoring".

One of these types focused on "participants absorbing information, often from examples or when attempting to understand a problem" [50]. This was aggregated with another form of Comprehension Monitoring that focused more on participants verbalizing that they did not understand their work, or explaining an idea to themselves. This second form was coded as *Monitoring Correctness* and will be discussed more in Section 4.2.1.

4.1.2 Scheduling. With 24 extracted passages from 14 different papers, the intentional optimization of time used to work was a frequently discussed skill in this corpus. In the context of the **Planning** skillset, *Scheduling* focuses on how students allocate time to work. It requires a student to assess the work they have to do and apply metacognitive knowledge to estimate how long they will need to complete it. They also must make judgments about the length and number of work sessions they will need to complete a task successfully.

Within this corpus, allocating time was often highlighted as important [2, 9, 14, 22, 55, 65, 67, 76], but few studies sought to define or explore the nuances of *Scheduling* outside of examining its indicators. For example, Arakawa et al. used indicators of *Scheduling* to identify students struggling to self-regulate, but did not expand into greater detail about *Scheduling* as a skill [2]. This skill is frequently highlighted as an important aspect of cognitive control. This corpus identified starting early and spacing as two potential indicators of *Scheduling*.

Starting early is summarized well in Zhang et al.'s 'Days Started Before Due' metric [104]. This indicator describes the difference between when a student starts an assignment and when it is due, and is a commonly used indicator of *Scheduling* [14, 41, 44, 104]. The metric 'days between starting and submitting the final product' is most frequently employed for longer-term assignments, typically with at least a week between assignment date and due date.

'Spacing' is used to describe a student distributing work over a period of time. Leppänen et al. summarize the educational psychology behind this behavior: "students who study the same set of material for the same overall time tend to perform better in tests if the studying is done in multiple spaced chunks instead of in a single session" [45]. As indicators, starting early and spacing are grounded in a significant body of general education research. Macan et al. noted in 1990 "allocating time properly or last-minute cramming for exams, have been frequently discussed as a source of stress and poor academic performance" [56].

4.1.3 Decomposing. Decomposition helps students determine an overall approach and facilitates better prioritization and allocation of time. The need to break down a task into smaller tasks or set intermediate goals for one's self was a frequently discussed skill. Indeed, there were 29 extracted passages from 19 different papers, making this skill second only to *Monitoring Correctness* (Section 4.2.1) in its frequency. As described previously, the term 'planning' meant different things in different studies. However, this idea of working to create sub-goals was the most common meaning of the term 'planning'.

There is some evidence in Education research to indicate that novices Decompose in a qualitatively different way from experts. Ho noted that experts tended to be more explicit and deliberate in how they decomposed a problem, while novices were not so structured in their behavior. This is not to say novices did not decompose. Instead, Ho noted novices still appear to have performed *some* problem decomposition based on analysis of the work produced [39], but appeared to do so internally. He divides these two styles of decomposition into 'explicit' forms, more frequently used by experts, and 'implicit' forms more typically employed by novices. Thus, creating observable artifacts may serve as both measurement and intervention.

Indicators for this skill often center around asking students to perform more explicit decomposition tasks. This was commonly done by looking for some document created as an artifact of a student's decomposition process. Many studies observe that while programming, students might outline code by making method skeletons or other design artifacts before starting to program [22, 82, 94]. The other common indicator was seeing students enumerate a list of "concepts, stages, or components'' [9]. This could be done by writing out a to-do list or verbalizing a set of goals [18, 22, 45, 46, 49, 65].

More recently, however, Christea et al. have examined how clickstream data from a learning management system could be used as an unobtrusive way to measure this construct. In their study, engagement with schedules, rubrics, or study guides on a learning management system as a proxy to measure self-regulation [11].

4.2 Monitoring Skills

Monitoring skills focus on different variables that students must maintain an awareness of. Within this corpus, we identified that many works commented on a student's need to monitor and validate the correctness of their work as well as their progress, given the allotted time. Another emergent theme we saw was that students should stay aware of their emotional state, or at least manage frustration and boredom as necessary to succeed.

Within Puustinen et al.'s framework, these skills would all be classified as forms of 'self-observation', which is part of the *performance* phase. The ways students act on that awareness were classified as forms of **Adaptation** and will be discussed in Section 4.3.

4.2.1 Monitoring Correctness. With 49 passages from 25 different papers, *Monitoring Correctness* is the most frequently discussed of our skills. Many studies call this idea "metacognitive accuracy'' and define it as "the relationship between confidence and correctness'' [86]. Lewis summarizes this form of self-awareness within the context of programming by asking "What parts of the code's behavior do and do not make sense?" [47] Both definitions capture the same idea: students need to regularly check in with themselves to make sure they understand what they are doing and are aware that their answers are correct.

In Think-Aloud studies, Validation could be as simple as a student explaining to themselves why a choice is right [47, 50, 65, 76, 82, 94]. Statements like: "this was the right loop condition because it halts at the end of the list'' [82] indicate a student is validating their choice to themselves. In a Think-Aloud study on Boolean logic, Herman et al. noted that many students did not notice mistakes in their work and only a handful validated their answers by translating the logic back to English [38]. Loksa and Ko looked for such behaviors when they observed students "...evaluating how well their implementation solves the problem, usually by testing and debugging'' [50].

Outside of settings where a researcher could observe Validation directly, studies have mentioned looking for evidence of testing code [9, 22, 50, 55, 94], though few studies do more than note its importance. Automated Assessment Tools (AATs) offer support when looking for testing behavior. Arakawa et al. used "fail[ing] to pass the same test case in three sequential commits & pushes to GitHub'' [2] as one of three indicators that a student might be struggling to self-regulate.

Another common way of observing this skill is to ask students to *Predict Confidence* immediately after finishing a test. While currently limited to just exams, these do a good job of measuring a student's *perception* of correctness and can be easily compared to the actual correctness of the work done. As these confidence predictions often take the form of a straightforward Likert-scale inventory, this form of measurement is easy to implement and has seen a great deal of study [12, 13, 43, 86, 91, 96]. Such predictions have a well-established basis in Psychology research. Hart reported in 1965 using "feeling-of-knowing" as a measure for what is and is not in memory [35], so it is unsurprising that this indicator shows consistent positive correlations with success across nearly every study that employs them.

Beyond 'feeling of knowing', *Monitoring Correctness* as a whole has been a well-known aspect of cognitive control for decades [66]. Baker and Brown described "monitoring the effectiveness of any attempted action" as an important metacognitive strategy in 1984 [3], and that is echoed in more recent research as well by Greene et al. [30]. However, Winne et al. have demonstrated that novices, overwhelmed with information, have difficulty with their ability to monitor their correctness effectively [98].

4.2.2 Monitoring Progress. A student's sense of where they are in solving a problem was another aspect of Monitoring discussed in this corpus. Loksa and Ko highlight 'Progress Monitoring' in their framework of self

regulation saying: "Programmers who explicitly monitor their progress toward solving a problem are more successful…The more learners monitor when a task is complete, the more successful they should be" [50]. Students need to have a metacognitive standard about how much time they need to devote to a piece of work and stay aware of their current pace. Students must then make adjustments to their schedule and behavior using that knowledge and awareness.

Prather et al. identified two key ways students misunderstood their progress through an assignment. Students associated seeing AAT feedback with being mostly finished with an assignment, even when there was still much work to be done. This assumption led students to have a "poor conception of location in the problem-solving process" [78]. Besides simply being unaware, Prather et al. also observed this hindered student's ability to question if they were on the right track as they had an "unwillingness to abandon a wrong solution due to a false sense of being nearly done" [78]. Thus, *Monitoring Progress* appropriately is important not only because it facilitates better allocation of future time, but also because it helps maintain other forms of awareness more effectively.

Ilves et al. scaffolded this form of awareness by showing students metrics about how they used their time compared to the class averages [41]. While this helped students practice staying aware of time, they did not use any behavioral indicators to judge how students improved over time.

4.2.3 Emotional Regulation. This skill is characterized by staying aware of one's emotional state and finding ways to succeed while taking one's emotional state into account. The passages that were assigned this code most frequently originated in the Motivated Strategies for Learning Questionnaire (MSLQ). Many statements in this inventory focus on students continuing to work through disinterest or boredom. This could be seen through statements like: "even when study materials are dull and uninteresting, I keep working until I finish'' [70]. Other questions more directly attribute disinterest to affect-related reasons like: "I often feel so lazy or bored when I study for this class that I quit before I finish what I planned to do'' [69]. In either case, the statements focused on maintaining energy to work when experiencing a de-motivating emotion.

There were also some questions which focused on more than just disinterest or boredom. These extended into resilience towards challenge and failure. The 1991 Effort Regulation subscale includes the statement "When course work is difficult, I give up or only study the easy parts'' [69]. Here, it is unclear if a student is experiencing anxiety, frustration, laziness, or some other emotion entirely. However, it does appear that some emotion is influencing their behavior.

Falkner et al.'s 2014 case study was the only work not using the MSLQ to discuss regulating emotions. In their work, they note "avoiding sources of anxiety'' was a form of personal management that students recognized was important for learning Computer Science. As this phrase had no additional context, it is unclear if this was brought up as a form of effective *Emotional Regulation* or ineffective.

No behavioral indicators or methods of measuring *Emotional Regulation* in action were discussed in any work.

4.3 Adaptation Skills

Whereas skills in the Monitoring skillset focus on maintaining an awareness, skills in the **Adaptation** category characterize how a student responds to such awareness. For example, if a student notices that they are stuck, then they might seek out help from an outside source. Similarly, if a student recognizes they are not making progress as they expected, they might decide to go to the library to work more effectively.

Within Puustinen et al.'s taxonomy, these skills would be classified as forms of 'self-control'. This, along with 'self-observation' is part of the *performance* phase.

4.3.1 Help-Seeking. Student use of outside resources (especially course staff) to help troubleshoot is frequently recognized as an important cognitive control skill [2, 16, 22, 29, 40, 55, 62, 82, 89, 105]. *Help-Seeking* requires students to observe and assess their specific situation in a number of ways. Doebling and Kazerouni note

"Academic Help-Seeking involves…identifying that help is needed to surmount the problem, deciding whether and from whom to solicit help, and finally obtaining and processing help'' [16].

Deciding who to solicit help from was a concept discussed by a few studies within this corpus.

A consistent sub-theme among works that discussed *Help-Seeking* was to focus in on a student's approach to *Help-Seeking* by noting what kinds of social assistance they sought or what kinds of questions they asked. Falkner et al. [22] discussed friends as a valuable resource for help, while Lyon et al. discussed getting help from TAs or instructors. The types of questions students asked was a point of discussion for Peteranetz et al. [68] who noted two different types of questions that students might ask. They write: "students asked questions to both advance their own understanding (high-level) or find out what the instructor wanted (low-level)" [68]. These two aspects of *Help-Seeking* both appeared to consider a student's approach to seeking help once they decided help was needed.

One (comparatively) new aspect of how a student sought help was explored in Prather et al.'s 2023 study on novice interactions with a large language model (LLM). As they note: " students will need better self-regulation skills to self-control their use of tools like [Google] Copilot to not develop an over-reliance on them - at least when they are freely available for use at the student's discretion. In fact, we hypothesize that over-reliance on tools like Copilot could possibly worsen a novice's metacognitive programming skills and behaviors." [79]. While the place of large language models within CS education is outside of the scope of this work, this aspect of help seeking will undoubtedly be the subject of a great deal of future study.

Lyon et al. [55] also considered the timing of when a student asks for help within their study. Among the self regulation strategies noted in their work, they included: "allowing failure or to reach the point of failure before looking for additional help" [55]. Their study was the only one in our corpus to touch on calculating when to ask for help. However, other work within the field has noted how correctly gauging when to seek help can be a big factor in successful *Help-Seeking*. Marwan et al. broadly categorized unproductive*Help-Seeking* into two forms: help over-use (or abuse) and help avoidance [59]. Seeking help too quickly (meaning that they have not sufficiently tried solving the problem on their own) or too frequently can reduce how well help is retained or how much is learned. Conversely, never seeking help or working independently for too long can both be frustrating for the student and can impact how much time a student has available to complete the remainder of their work.

While many studies recognize this to be a key aspect of cognitive control, only a few studies have examined patterns in *Help-Seeking* behaviors. Ren et al. created a formalized "Design Recipe'' to help understand what happens in TA-run office hours. Students and TAs filled out a form after interacting to characterize what the student needed help with and what steps of the problem-solving process they addressed together. Students most frequently sought help with analyzing a task, translating function outlines into code, and testing code [82].

Doebling and Kazerouni focused on characterizing the *Help-Seeking* process more broadly. They found that students considered all resources useful, but tended to progress from informal and easily accessed forms of help to progressively more formal sources when they encountered a difficult problem. This commonly took the form of first using online resources, then asking peers, and only then seeking help from course staff if their question had not been answered [16].

Doebling and Kazerouni also recognized how identity or looking foolish influenced help avoidance: "Computing students can be prone to comparing themselves to peers and taking actions to affirm their sense of belonging in computing. One such action might be help-avoidance in order to affirm one's identity as a computer scientist or programmer.'' [16]. This issue does not seem to be limited to *Help-Seeking* in Computer Science courses either. For instance, Won et al. note that "…students may also associate seeking help with personal inadequacy and embarrassment and perceive it as a sign of weakness and a threat to their self-worth" [95]. Similarly Fong et al. note that those that avoid help experience less "emotional support and social efficacy" with instructors [26]. While this corpus of works did not delve deeply into this idea, these observations seem to indicate that identity is a critical variable to effective *Help-Seeking*.

With this in mind, Önder et al. also sought to find ways to measure help-seeking behaviors unobtrusively. Time within a Learning Management System (LMS), posting in a discussion forum, and engaging with assignments in an LMS were, among other things, used as proxy indicators to help identify when a student might be seeking help [62].

4.3.2 Reducing Distractions. Students who are skilled in *Reducing Distractions* are cognizant of how they study best and can assess and change environments in order to suit their needs. The MSLQ Time and Resource Management subscale includes several statements about a student making intentional choices about their study environment like: "I usually sit in a place where I can concentrate on my course" [69]. In a qualitative analysis of student self-reflection, Prather et al. reported Environmental Restructuring was a consistent theme [76]. Falkner et al. similarly reported "reducing distractions" among the themes from their qualitative analysis [22].

With only four passages discussing this skill, we could not identify any discernible trends or specific behaviors that could allow an outside researcher to reasonably infer that a student is intentionally using this skill. For example, while knowing when a student has gone to the library to study could be track-able, this in itself does not demonstrate cognitive control without knowing intent. Instead, it is likely easier to observe this skill in a more simulated setting like a laboratory where a researcher could ask about a student's reasoning for choosing to study in a certain location.

However, *Reducing Distractions* is a key aspect of cognitive control within more general education literature. Ley and Young reported that 'environmental structuring' was highly related to effective Monitoring and noted this as the "strongest difference between college students who used an instructional self-monitoring protocol and those who did not" [48]. Additionally, besides being a significant theme of Pintrich's MSLQ inventory, it is also a key SRL strategy in the Zimmerman and Martinez-Ponz taxonomy [28, 107]. This theoretical foundation seems to indicate that *Reducing Distractions* could be a promising avenue for future study in CS classrooms.

4.4 Reflection Skills

The final phase of Puustinen et al.'s taxonomy is the appraisal phase wherein students consider their process and the work they produced in aggregate and resolve to make changes for the future [80].

With twelve total passages in this corpus discussing **Reflection**, this skillset is still being explored to its fullest. Indeed, **Reflection** was identified in many papers with little context, saying only things like "self-evaluation" as an aspect of cognitive control [55, 76].

However, this skill does appear to be gaining a some increased interest in the research community more recently. 3 out of the 12 papers from that explored self-reflection did so in 2023. Zahn et al. used reflective writing as a way to "help students self-identify the struggles they are encountering and areas for improvement" [103].

One component of the Winne-Hadwin model is the "COPES" schema which details how students evaluate success based on a set of standards [97]. While infrequent, the research team did notice two trends in what standards students applied when reflecting on past work. Sometimes researchers focused on how students reflected on assignment feedback [2, 11] or, more generally, the expectations of the class [40, 69]. These standards required *Reflection using External Standards*, which come from sources external to the student. For example, the MSLQ includes the statement: "I try to change the way I study in order to fit the course requirements and instructor's teaching style" [69]. This is contrasted with *Reflection using Internal Standards* which focused on how a student evaluated their work against their own, more internally facing, standards. For example, Stephenson et al. [92] report an intervention where students had to complete an exam wrapper that asked them to "reflect how they prepared for the exam, the kinds of errors that they made while answering the exam questions, and the changes that they would make for next time". In this situation, a student is reflecting on how they performed compared to their own expectations of themselves, rather than a class-set or instructor-set standard.

Two studies identifed ways to examine reflection in action among students. Hundhausen et al. identified reflective writing within chat rooms of group project work within a software development class to unobtrusively examine behavior [40]. Christea et al. also sought to identify reflective behaviors unobtrusively, through digital traces of engagement with an LMS. They identified accessing assignments after a deadline, examining the gradebook, and engaging with quizzes focused on feedback [11] as useful indicators for an educator to use when trying to identify reflection in action.

However, **Reflection** likely will always be a challenging skill to observe. Like *Reducing Distractions*, using just behavior to identify if a student is engaging with some **Reflection** skill cannot tell us much about their intent. Thus, making inferences between what a student does and what's going on in their head is difficult. Still, since reflecting is such a significant part of so many theoretical frameworks of cognitive control, any successful study to expose how a student goes about thinking back on their work could contribute to a better understanding of cognitive control as a whole.

4.5 Knowledge Building

Knowledge Building was not classified within any of the previous four skillsets as it could occur in any of Puustinen et al.'s three phases [80].

This skill encompasses the ways a student independently builds knowledge outside of what is needed for class credit. This skill was almost always discussed in the context of an indicator, rather than as a skill itself. Seeing students *Practice* work on non-credit exercises [22, 70, 76, 104] was the most frequent indicator used. Most studies did not discuss this beyond noticing that practicing was important for success and an aspect of cognitive control. For example, Prather et al. [76] simply include "practice coding" as one of the self-regulatory skills important for CS success. Seeing students investigate, or "build knowledge by exploring other resources" as Falkner et al. put it [22], was also occasionally brought up as an indicator [68].

Similar to *Reducing Distractions*, *Knowledge Building* presents some difficulties for further study as it is difficult to measure intent. For example, an outside observer watches a student search for information on something outside of the course. This could be demonstrating investigation, or it could just as likely be demonstrating that a student has fundamentally misinterpreted what they need to be doing. Without the knowledge that a student is intentionally searching for information outside of what is required for a course, one cannot make an inference that the behavior indicates *Knowledge Building*.

4.6 Procrastination

Procrastination can be defined as intentionally delaying work in spite of an awareness of the negative consequences of the delay. Since it does not fit within our definition of skill, procrastination was categorized as a "general indicator" and is not within another one of our skillsets. Within the works examined in this review, procrastination was regularly framed as a signifier of a difficulty regulating, rather than a cognitive control skill [2, 85, 91, 102, 104]. However, opinions differed on which specific skill difficulties were at the heart of the problem.

Procrastination is often thought of as difficulty with setting aside appropriate time. Arakawa et al. discuss procrastination as an underestimation of time in their analysis, noting some students habitually waited until just before a deadline while others realized that the time they set aside was insufficient once they were too close to the deadline [2]. Zhang et al. measured procrastination as the number of days before a deadline an assignment was started [104].

However, other papers in this corpus attributed procrastination to difficulty with other skills. Shaffer and Kazerouni [85] took a different approach and focused on both an awareness of time and an ability to decompose abstract projects. In their study, the authors introduced project milestones that were intermediate due dates for specific components of an overall project. These milestone helped create more tangible deadlines with straightforward

work because "a task whose outcomes are farther in the future are more likely to invite procrastination". This intervention demonstrated Decomposition to students and provided a structured way to learn about *Monitoring Progress*.

Martin et al. [19] examined procrastination in other ways by examining interventions each seeking to scaffold a different skill. Their first intervention focused on reflective skills and asked students to write a few sentences on their use of time in the prior assignment before starting the next project. Their second intervention had students develop their *Scheduling* ability by creating a written schedule sheet. Here, the aim was to help students "form, express, manage, and track smaller-scale deadlines" [19]. Third, automated email alerts detailing how far a student had progressed compared to their peers was introduced to help students Monitor Progress. While not explicitly conceived with this in mind, these three interventions focus on the three common phases of self regulation. Schedule sheets represent a way to help reduce procrastination in the *preparatory* phase, automated emails help students self-observe in the *performance* phase, and written reflections train students to *appraise* their work.

Pereira and Díaz [67] recognize three different types of procrastination that are similar to the phases outlined by Puustinen et al. – though with different names. Before starting, students may have "difficulty planning and prioritizing tasks". While working, they may delay work because of an inability to focus and avoid distraction. After working, a student's choice to start working on similar tasks may be influenced by self-efficacy.

5 Discussion of RQ1: Coverage of Identified Skills

Our first research question was: *What aspects of cognitive-control skills have been examined in post-secondary computer science education research, and what behaviors have been used to indicate skills?* Thus, we seek to provide a summary of what areas within cognitive control have been targeted previously. This question also seeks to identify ways students demonstrate cognitive control. This is useful to know since, in order to assess an effective intervention, one needs to recognize when a student has improved. We answer our first research question through the data summarized in Tables 3 and 4. Our review identified 11 skills and 10 observable behaviors that can act as indicators of those skills in students. These skills were diverse in that they address what students do before, during, and after completing a task.

Areas of focus — While this list shows cognitive control work has examined a breadth of skills, there is a clear focus in this corpus on only a few of these. *Monitoring Correctness*, *Decomposing*, and *Scheduling* were the most discussed, while *Emotional Regulation*, *Reducing Distractions*, and **Reflection** were the least discussed. Passages highlighting *Emotional Regulation* or *Reducing Distractions* came frequently from works using the MSLQ. However, outside of this general education inventory, they were not often discussed in the context of CS directly. This means the nine studies which used the MSLQ ended up being the vast majority of works to measure things like *Emotional Regulation* or *Reducing Distractions* [8, 9, 18, 33, 45, 46, 49, 76, 90], albeit indirectly. Overall, there appear to be noticeably fewer studies examining how students respond and reflect as compared to how they plan and monitor. This observation appears to extend beyond the works in this corpus into other literature reviews on self regulation and metacognition [28, 87]. However, we are hopeful the research community has already begun to change in this respect, given that one quarter of the papers in this corpus that examined reflection did so within 2023.

Even if change is not immediately around the corner, this is not necessarily a weakness of the current state of CER, as there is still so much we do not know about even the most well-studied skills. However, *diversifying which skills we examine could better inform instructors about what needs the most instruction in future.*

Help-Seeking and Procrastination — *Help-Seeking* and procrastination are two areas presenting special challenges, and these problems may need to be addressed in future study. Both terms describe complex and multivariate aspects of cognitive control, which makes comparing results of different studies especially difficult.

As the field advances, the CER community would benefit from developing shared vocabulary that helps to tease apart the specific aspects of these broad constructs that are being studied in a particular work.

Help-Seeking encompasses a wide variety of judgments regarding whether one needs help, what question to ask, when to ask it, what resources to use, and what to do with the help once received [16]. These judgments are each qualitatively different from the others, and students could have different levels of proficiency at each. For example, a student could recognize the right time to ask a question, but struggle to form a question that will help them effectively learn. Similarly, procrastination can indicate a failure before, during, or after working on an assignment [104]. This makes it difficult to compare the results of different interventions for these areas since researchers are typically intervening only on specific components of a whole. Shaffer and Kazerouni [85] created an intervention seeking to reduce procrastination by creating milestones that forced students to decompose, while Pereira and Díaz examined the efficacy of automated reminders so as to scaffold *Monitoring Progress* [67]. These studies are all seeking to reduce procrastination. Yet there are so many different aspects of this issue to intervene on. A student might quickly Decompose a task, but end up Procrastinating when they do not stay aware of how much time they are spending on a part of the problem. Alternately, a student might have a strong track record of managing time effectively but then poorly allocate their time due to difficulties with *Decomposing* the task. Thus, all of these studies seek to intervene on procrastination, but approach the problem in such different ways that it is difficult to compare Shaffer and Kazerouni's work to that of Pereira and Díaz. Thus, future work may benefit from attempting to break *Help-Seeking* and procrastination into smaller components.

It is our hope that researchers will continue to refine and explore these areas of self-regulation, as they appear to be increasingly important in the world today. While only briefly touched upon within our corpus of works, the presence of Large Language Models (LLMs) within our lives makes these two areas of self-regulation all the more valuable to explore. Asking an LLM for help involves none of the social stigma of asking a peer or course staff member for help, yet over-reliance on such tools are also likely unproductive for actual learning [79]. Similarly, a LLM's ability to assist might encourage delaying the start of work since such a tool either heavily speeds up the programming process or can provide an answer without a student needing to type any code to begin with.

6 Relating Skills to Success

In this section, we discuss the degree to which skills studied in this corpus have a strong or promising connection to academic success.

Table 6 provides an overview of the 30 studies that related a skill to academic success in some way. We use the word 'observe' to mean that a study noted a relationship between a skill and success, but did not present any statistical analysis. This means results from qualitative studies will be reported as observing a relationship between a skill and success. Additionally, there are several studies that reported differences in average or median score between control and experimental populations, but did not test to see if the difference is statically significant. These studies were also categorized as reporting an 'observed' relationship. This is not to say these observed relationships are not valuable. Indeed, these studies help point towards promising new directions for future work. Instead, we use the word observed in this way to communicate a difference in the generalizability of the results found.

MSLQ. As described in Section 4, the various forms of the MSLQ inventory often aggregate several skills together to form one holistic score that describes a student's ability to self-regulate. Therefore, Table 6 does not report on any studies that used this inventory to measure cognitive control. Instead, Table 7 lists each study, the version and subscales of the inventory they used to measure self regulation, and whether or not the scores shared a statically significant relationship with academic success.

Skill	Statistically Significant Relationship With Success	No Significant Relationship With Success	Observed Relationship With Success	Unique Studies Relating To Success
Planning (skillset)		[88]		1
Task Analysis (skill)	[11, 13, 50]		[77, 78]	5
Scheduling (skill)	[1, 10, 14, 41, 44, 104]		[2, 44, 55]	9
Decomposing (skill)	[11, 50, 85, 94, 105]			5
Monitoring (skillset)	[105]			
Monitoring Correctness (skill)	[12, 43, 50, 94]		[2, 78, 86, 91, 96]	9
Monitoring Progress (skill)	[50, 94]		[78]	3
Emotional Regulation (skill)				0
Adaptation (skillset)			[10]	1
Help Seeking (skill)			[55]	1
Reducing Distractions (skill)				0
Reflection (skillset)		[55]		1
Reflection - Reflection using External Standards (skill)		[92]		1
Reflection - Reflection using Internal standards (skill)		[40]		1
Knowledge Building (skill)		[104]		1
[Reducing] Procrastination	[102]			1

Table 6. Studies Which Relate a Skill with Academic Success Metrics (Does Not Include Papers Using the MSLQ)

Reference	MSLQ Version	Subscales Used	Statistically Significant Relationship With Success
[18]	MSLQ -1991	Metacognitive Self-Regulation Effort Regulation	Unreported
[9]	MSLQ -Columbia	Planning Monitoring Study Method Regulation	No
[76]	MSLQ -1991	Metacognitive Self-Regulation (3 questions)	Yes
[8]	MSLQ-1990	Metacognitive Self-Regulation	No
[49]	MSLQ -1991	Metacognitive Self-Regulation	No
[90]	MSLQ-1990	Metacognitive Self-Regulation	No
[33]	MSLQ-1991	Metacognitive Self-Regulation	Yes (indirect)

Table 7. Results relating MSLQ Inventory Scores to Academic Success

Three studies reported statistically significant results regarding MSLQ scores and academic success metrics. First, Duvall et al. found a moderate positive correlation between four individual statements from the Metacognitive-Self Regulation (MSR) subscale and concept inventory scores, but do not note whether the relationship is significant [18]. Second, Prather et al. reported a weak but significant positive correlation between success metrics (like test score and project score) with the sum of three MSLQ questions from the MSR subscale [76]. Third, Han and Ellis clustered student data based on, among other things, MSLQ scores and course grades. They reported a statistically significant difference between their clusters using a one-way ANOVA test, indirectly demonstrating a relationship [33].

6.1 Skills With a Demonstrated Relationship to Success

6.1.1 Monitoring Correctness. As the most frequently discussed skill in this corpus, it is perhaps no surprise that *Monitoring Correctness* bears a consistent relationship with success.

Predicting Confidence is measured in several works within this corpus. Predicting Confidence involves either asking students to predict which questions they answered correctly on an exam or asking students to predict their

overall exam score. These predictions of correctness or score are then interpreted as predictions of confidence. Stephens-Martinez writes: "Generally, our students are reasonably accurate within 10% of their actual grade", and that observation seems to be consistent with papers both within the corpus [12, 86, 91, 96] and more generally as well [34]. Most studies also report that lower-performing students are worse at these predictions than middle or high performers [12, 43, 91], and they also tend to be overconfident.

While these predictions are a useful tool for measuring a student's ability to Monitor Correctness, this approach does come with some limitations. Denny et al. [12] and Stephens-Martinez [91] both note test predictions did not significantly change in accuracy throughout the course of a semester, implying they are not a good catalyst for behavioral change. Additionally, while they do appear to be an effective tool for measuring *Monitoring Correctness*, there could potentially be a ceiling to their efficacy for even this purpose. If students are already fairly accurate at Predicting their Confidence, then an intervention seeking to improve their ability to Monitor Correctness can do little to increase how accurate they are at the specific behavior of Predicting Confidence.

Validating is another approach to observing *Monitoring Correctness*, though evidence for this indicator's relationship to success is more conflicted. In their Think-Aloud study, Sudol-DeLyser [94] examined how self-explanation at different levels of abstraction differed among higher and lower performers. They found no significant differences between their success metric (number of submissions to an AAT) and different types of self-explanation. However, they did find a significant difference in the types of self-explanation that High Performers engaged in versus Low Performers. High performers more frequently engaged in self-explanation that related disparate elements together or to the overall algorithm. They note: "This would indicate that higher proficiency students are more likely to use abstract statements during the self-explanation process" [94]. In Loksa and Ko's Think-Aloud study, 'self-explanation' was a code used to describe when students verbalized "an account of why a decision was correct" [50]. In their multiple linear regression of CS2 students, increased frequency of these statements significantly corresponded to an *increase* in errors in code. That more self-explaining correlated with more errors seems to indicate the behavior is more frequent in students who are struggling. This appears to conflict with Sudol-DeLyser's findings, but Sudol-DeLyser used submissions to an AAT as a success metric while Loksa and Ko evaluated code by looking for errors.

As *Monitoring Correctness* seems to already have a clear relationship with success, we would expect validating behaviors that demonstrate that awareness to also have a close relationship with success. However, as only these two studies examined validating behaviors in this way, we can only note further examination into this behavior is likely needed.

In Loksa and Ko's Think-Aloud study [50], the authors had an opportunity to be in the room as students verbalized their process of *Monitoring Correctness*. Statements where students reflected about their "the understanding of code or problem prompts" [50] were coded as Comprehension Monitoring. Loksa and Ko's 'Comprehension Monitoring' category combines what we have defined as *Monitoring Correctness* and *Task Analysis*. In their linear regression of CS2 students, the frequency of this code significantly related to a decrease in code errors. Since this code aggregates two different skills from this review, it is unclear how significant this finding is when relating *Monitoring Correctness* to success.

Overall, it appears that there is a correlation between higher performing students and an awareness of correctness, but this corpus largely focused on one method that does not effectively scaffold improving the skill of *Monitoring Correctness*.

6.1.2 Scheduling. Several studies relate indicators of *Scheduling* to success. Three studies showed a significant correlation between starting early and academic success, one showed a statically significant relationship between spacing and success, and one used both indicators together in an intervention that particularly helped weaker students.

Starting early was significantly related to academic success in three papers from the corpus [14, 44, 104]. Denny et al. [14] used optional early feedback deadlines as an incentive to help encourage starting early. They found students who engaged with early deadlines earned significantly better grades compared to those who did not. However, the degree to which starting early is important is not yet clear. Zhang et al.et al.[104] saw weak but significant correlations between starting early and homework grades overall. They also found a "weak but statistically significant correlation between average DSBD [days started before due] over the homeworks completed before the first exam'' and that first exam score [104]. However, they could not replicate these findings with starting homework early before the second exam and that exam's grade. Leinonen et al. noted a "clear trend" where students who started earlier on problem sets outperformed those who did not, but also note: "even of those who start one day before the deadline, most will get a 5, i.e. the best grade available. Only for those who start on the very last day, the day of the deadline, the most likely grade is 0 or 1" [44]. For these homework assignments, the key was not how early a student started, but that they left at least a day to complete the work. Thus, it may be valuable for future work to investigate how early is early enough for different types of assignments.

Overall, these studies show evidence that starting early has a significant correlation with academic success. However, the degree to which it is important is still not yet clear.

Spacing refers to the process of distributing smaller time blocks throughout the course of an assignment. In this corpus, only one paper related this skill to academic success, but the relationship they found appears promising. Spacing is in contrast to 'cramming' (allocating one large block for all work) [45], typically right before a due date. Chung and Hsiao [10] studied the consistency of students using an optional practice platform. Of the students who interacted with the system, those who consistently used it week-to-week outperformed those who used it only right before an exam or inconsistently. However, lower-performing students who regularly engaged with the platform did find value in this cramming strategy. When examining the behaviors of just low-performing students who consistently used the platform, they note students who transitioned to more cramming patterns outperformed those who did not.

Starting early and spacing were used together in one study [41]. Ilves et al. gathered data on starting early and spacing behaviors compared to the class average. This was presented to students in two different ways: textually and through a graphic visualization. They noted among the high performing students that the interventions made little difference, but among the lowest third of the class, the graphical presentation had a statically significant relationship with exercise performance compared to the control [41]. In this intervention, we cannot determine whether one indicator contributed more than the other, but increasing overall awareness seems to have helped.

6.2 Skills with a Promising Relationship to Success

Skills discussed in this section have some evidence that they hold a relationship with success, but more study is needed before we can say for sure.

6.2.1 Task Analysis. *Task Analysis* has been observed to be important, but statistically significant results have not replicated these findings. Four studies sought to connect *Task Analysis* with academic success [13, 50, 77, 78].

As mentioned in Section 6.1.1, Loksa and Ko's Comprehension Monitoring code encompassed both *Task Analysis* and *Monitoring Correctness*. This finding, therefore, indicates a statically significant relationship between academic success and this pair of skills. However, it is unclear how direct this relationship is.

In 2018, Prather et al. performed a Think-Aloud study looking for common metacognitive difficulties that students face when using an Automated Assessment Tool (AAT). They observed that proficiency at forming a correct conceptual model was one of the biggest differences between students who successfully completed the assignment and those who did not. The authors summarize: "many of the students who did not complete the quiz read the prompt (often briefly) and jumped directly to coding…This proved disastrous for them as they wandered aimlessly, seeming to hope they would eventually stumble on a solution" [78]. This study focused on

characterizing metacognitive difficulties that students encountered when interacting with an AAT, and part of the conclusion was that these tools need to provide "implicit support of metacognitive awareness" [78]. One area they suggest for future work is to create software that indicates whether a student understands what they are doing before beginning to code.

Prather et al. and Denny et al. both sought to create interventions to do just that. All students were given a programming problem but students in the experimental group had to first "solve a randomly generated test case after reading the problem prompt" [77] before they could begin coding. The major differences between these two studies were the methodologies and population sizes used. Prather et al. conducted a Think-Aloud study on a small population (n=38) where all students were observed interacting with the programming problem and intervention. They observed "the experimental group had a higher completion rate, faster time, and fewer attempts required to complete [the assignment]" [77] [77].

Denny et al. [13] conducted a larger scale (n=976) quantitative follow-up study where students in the experimental branch were required to solve a test case before starting a homework assignment. They reported that students in the experimental group had a statically significant reduction in logic errors compared to the control. However there was no significant difference in completion time, success rate, or number of submissions to the AAT. These findings seem to largely contradict what was observed by the previous two Think-Aloud studies, but Denny et al. do note several limitations in their study and recognize the need to replicate the experiment "within a more structured laboratory environment" [13] to be more confident in their results.

Prather et al.'s 2018 study [78] demonstrated the importance of *Task Analysis* for students in CS, noting that jumping into coding before starting "proved disastrous" for students who did not fully read the prompt. The subsequent two studies then tried to analyze the effect of preventing this specific behavior by not allowing students to code until they had solved a test case, but these had mixed results. At a minimum, a third follow-up study would be needed to validate whether preventing students from jumping into coding helps scaffold this skill. There is also further work to be done in this area to assess what kinds of assignments this intervention works for. Is there value in such test cases for coding assignments which are larger in scale and complexity? What about more theoretical assignments?

There is also more work to be done assessing other facets of this skill. As stated in Section 4.1.1, correct understanding of the task is desirable, but *Task Analysis* requires that students know how to intentionally take time to internalize requirements before starting. Asking students to solve test cases before starting work is one approach to encouraging this behavior, but it is not the only way to do this. The efficacy of encouraging students to annotate the problem statement, for example, could be another important avenue for future investigation into this skill.

6.2.2 Decomposing. While Decomposition is one of the most frequently discussed skills in this corpus, it has seen limited study relating it to academic success. Out of the eighteen studies that discuss this skill, only two relate Decomposition to success. However, both of these studies found promising evidence. In their Think-Aloud study [50], Loksa and Ko's regression model relates the frequency of their categories among students to errors in the code they wrote. They found that frequency of statements related to starting or revisiting sub-goals significantly related to fewer errors among more experienced students. Shaffer and Kazerouni created an intervention where students had to meet mandatory intermediate deadlines where different sub-goals of a longer-term project would be due. While they sought to reduce procrastination on programming projects, their implementation focused on delays "stemming from difficulties with decomposing a large engineering task" [85]. They report that while pass, fail, and withdraw rates stayed the same, students in the intervention group overall produced code that was significantly more correct than students in the control. As a result, the grade distribution changed with the effect of adding a letter grade to the middle third of the class.

Given these results and the fact that Decomposition is so widely acknowledged to be an important part of cognitive control, this skill holds great promise for future research. It is especially notable that none of the commonly discussed indicators for this skill in Section 4.1.3 (enumerating and outlining) were studied in relation to academic success in any work in this corpus.

7 Discussion of RQ2: Common Patterns among Studies

Our second research question was: *Which cognitive-control skills (or associated behaviors) have a demonstrated relationship with post-secondary student success in computer science education?* In this section, we highlight three major takeaways from the analysis of Section 6. First, it is likely that the field of cognitive control study would benefit from greater diversity in what skills we study and how we approach that study. Second, cognitive control interventions appear to impact students at different performance levels differently. Third, the MSLQ may need some additional validation within a CS classroom as it was a weak predictor of success within this corpus.

1. Diversifying Cognitive Control Study: As discussed in Section 5, diversifying what we study could be beneficial. As Table 6 shows, there are 24 studies that relate any skill identified in this work to success. 16 of those studies (66.66%) relate *Monitoring Correctness* or *Scheduling* with success. Other aspects of cognitive control, such as **Knowledge Building**, are still things the research community are just beginning to explore. Without further examination we cannot say for sure. Thus, future research into what skills most directly impact academic success could help educators pinpoint what parts of cognitive control to intervene on.

Additionally, future work might find it valuable to diversify the methods we use to examine cognitive control skills. For example, there are five studies in this corpus that examine *Monitoring Correctness* by asking students to Predict Confidence. *Monitoring Correctness* overall is widely agreed to be valuable, but confidence predictions do not appear to be a causative skill that improves performance. Additionally, the context of these confidence predictions was limited to exams. Extending these types of questions to other types of assignments, like programming homeworks or larger-scale projects could be a valuable new space for this form of measurement. Predicting Confidence is just one example of the need for diversity. Scheduling was most often measured by noting when students started assignments. Studies examining *Task Analysis* focused mostly on discouraging rushing.

Along with different methods, it might be helpful to broaden the indicators used to capture cognitive control. There was a trend among studies in this corpus to use indicators that exemplify how educators *want* students to act. This is not to say such indicators are unhelpful for research, just that they cannot describe cognitive control in isolation. For example, it is easy to make the inference that a student who Starts Early or Spaces out their work is intentionally *Scheduling*. Conversely, the student who started one day before the due date might have intentionally allocated the right amount of time, or could be scrambling to finish. As outside observers we do not have the means of differentiating from a timestamp alone. However, a student who allocates the last six hours before a deadline to work because they assessed their work and Scheduled accordingly is still demonstrating cognitive control. Starting early and spacing also do not take the greater context of a student's situation into account. For example, *Scheduling* behaviors might look different for a student with many other commitments (like a job or a heavy course load) than it does for others.

2. Cognitive Control for Classroom Equity: Within this corpus it became clear that cognitive control interventions might be more equitable than equal. There were several studies that found interventions helped students who were struggling to succeed more dramatically than students who were already succeeding. This observation makes intuitive sense as those that have stronger cognitive control skills are more likely to succeed in any context. We highlight it here, though, as students who are struggling tended to respond to cognitive control interventions in qualitatively and quantitatively different ways that might not be visible when evaluating a class as a whole.

Ilves et al. [41] sought to scaffold *Scheduling* by creating an intervention where students in the experimental branch saw visualizations of how their work patterns differed from class averages. When comparing overall populations, they reported no significant differences between control and experimental populations. However, when they stratified their students into thirds based on performance, it was clear that students in the top tertile tended to do well with or without an intervention while students in the lowest tertile were significantly more successful with an intervention. Other studies have also noticed that students at different performance levels respond differently to cognitive control intervention. Denny et al. [14] noted that struggling students saw more dramatic improvement than their high performing peers. Shaffer and Kazerouni found that their intervention helped students in the middle more dramatically than high performing or low performing students [85]. Thus, future work may find it valuable to measure the impact of a cognitive control intervention on different brackets of student performance as well as the impact on the class overall.

Chung and Hsiao [10] also observed a situation where students demonstrated cognitive control in a qualitatively different way. In this study, the authors examined how different patterns of engagement with an optional practice platform related to academic success. When looking at the class overall, Chung and Hsiao reported a moderately positive correlation between consistent engagement and exam score. However, they also examined transitions in engagement patterns among different populations of students, including students who failed the first exam of the class who consistently practiced. Among this specific population, those that moved from a consistent pattern of engagement to a cramming pattern (practicing just before an exam) outperformed their peers in subsequent exam scores. The authors hypothesize that this bracket of lower-performing students, those that engage consistently "…do not practice effectively…having a high error rate or gaming the system for memorizing more questions and answers" [10]. In this case, switching to a cramming approach might have helped these students take a more thoughtful approach. We highlight Chung and Hsiao's results here as it shows a way in which the successful approach for the class overall and the successful approach for lower-performing students did not align. This was the only study to report such a finding within our corpus, but examining how different performance levels of students demonstrate cognitive control could be an interesting avenue for future work to explore.

3. The MSLQ and Academic Success: MSLQ scores do not seem to reliably predict success among works in this corpus. While certain statements from the inventory have been discussed in previous sections, it is important to note that most studies in this corpus take the aggregate score of all of those questions. No significant correlations were found between aggregate scores and success [8, 9, 90], nor did any study conducting a regression analysis note the MSLQ as a significant factor [8, 90]. Lishinski et al. conducted a path analysis, but they also did not see a significant relationship between MSLQ scores and any success metric [49]. Han and Ellis were able to demonstrate a significant relationship between clusters, which included MSLQ scores among other variables, making it difficult to understand the inventory's role in their analysis.

While there is a little evidence that aggregating a smaller number of questions correlates with success [18, 76], there are a few pieces of evidence that indicate this inventory may not be useful for modern Computer Science students. Within the study of education generally, Broadbent and Poon noted that the MSLQ: "…may not capture the construct of online learner self-regulation as accurately as online-focused, validated measures" [6]. This makes some sense since the most recent inventory was published in 1991 — meaning it predates the adoption of Google as a search engine by a decade. Thus it is possible this inventory is not asking questions that accurately measure the study strategies of modern CS students. Within CS education, Prather et al. [75] have also critiqued self-reporting as a method of measuring cognitive control. They note: "Self-report measurements of cognitive control, such as the MSLQ, often measure what students think they do, rather than what they actually do" [75].

8 Conclusion

While cognitive control starts in early childhood, students greatly benefit from direct instruction even at the post-secondary level [23, 50, 53, 54, 61, 84]. However, it is not yet clear which cognitive control skills are most valuable to intervene on or how students demonstrate those skills in their behavior.

To better examine what skills within cognitive control are valuable to CS students, we conducted a review of cognitive control literature in post-secondary CS education. Our goal was to identify skills that the CS education community has studied, what behaviors they use to observe them, and what connections they have drawn between those skills and academic success metrics. To do this we reviewed works from the ACM Full Guide to Computing Literature online library. We constructed a corpus of 51 works that matched our eligibility criteria. We identified 11 cognitive control skills and 10 behavioral indicators used to observe those skills, all of which are grounded in educational theory.

Among the studies examined here, we found that planning skills have been relatively frequently studied, as has *Monitoring Correctness*. Other forms of **Monitoring**, **Adaptation**, and **Reflection** have received less attention. We observed that *Monitoring Correctness* and *Scheduling* had the greatest number of papers relating a skill to academic success, but studies in this corpus tend to measure them in particular ways that do not necessarily capture the full picture of either skill.

Overall, future work may benefit from diversifying both what and how we approach the study of cognitive control. Some skills, like *Reducing Distractions* or *Emotional Regulation*, have firm theoretical backing but were rarely the focus of studies in this corpus. Other skills, like *Task Analysis* and Decomposition, appear promising, but need more evidence to conclusively relate them to success. Even frequently studied skills, like *Monitoring Correctness* or *Scheduling*, could benefit from novel approaches to better understand how they work.

From conducting this review, it is clear that even among the most frequently studied aspects of cognitive control, there is still much to explore. It is our hope that this review will inform and spur future work so that students will better learn how to effectively learn. The works in this review identified a wide variety of skills a student will use before, during, and after working. Still, finding conclusive evidence that many of these skills related to success in a meaningful way was not possible. In many cases skills, such as *Monitoring Progress*, have simply not seen enough focused study within this corpus to identify trends.

We echo recommendations by Prather et al. and Loksa et al. [52, 75] that precise language to discuss what we as a community mean when we talk about cognitive control can only help us better understand this construct. We also need greater precision in our language about individual constructs. For *Help-Seeking*, procrastination, or even the term 'planning', it may be helpful to come to some consensus about what precisely these words mean and what is within their scope. Getting more precise about these terms could also help us mentally re-frame these aspects of student learning, helping us better identify ways to support it.

More data points can help the research community better come to know this aspect of the student learning experience. We hope that within this review, we have offered some promising directions for that future research. From the skills we study, to the behavioral indicators we look for, to the very ways we construct our studies, there appears to be ample opportunity for future work to build off of the foundations of existing work. Novel approaches or frameworks for understanding cognitive control may not lead anywhere themselves, but could help clarify what does and does not work about our existing ideas on this subject.

Overall, it appears that the study of cognitive control currently offers a great deal of space for creativity and new ideas in studying student learning within CS. The field of Learning Analytics is starting to be leveraged to study these skills unobtrusively and even help students learn how to be more aware of their own habits. Thus, with a lot of reasons to try something new and a lot of new tools to try, it appears that it is a particularly exciting time to study cognitive control.

A Final Corpus

Citation	Title	Authors	Year	Published in
[7]	Pencil Puzzles for Introductory Computer Science: an Experience- and Gender-Neutral Context	Butler, Bezakova, and Fluet	2017	SIGCSE
[92]	Exam Wrappers: Not a Silver Bullet	Stephenson, Craig, Zingaro, Horton, Heap, and Huynh	2017	SIGCSE
[55]	Student Challenges, Strategies, and Learning Within the Data Mine Learning Community	Lyon, Jaiswal, Magana, Gundlach, and Ward	2021	FIE
[18]	Improving Content Learning and Student Perceptions in CS1 with Scrumage	Duvall, Spurlock, Hutchings, and Duvall	2021	SIGCSE
[91]	A Study of the Relationship Between a CS1 Student's Gender and Performance Versus Gauging Understanding and Study Tactics	Stephens-Martinez	2021	SIGCSE
[16]	Patterns of Academic Help-Seeking in Undergraduate Computing Students	Doebling and Kazerouni	2021	Koli Calling
[68]	Computational Creativity Exercises: An Avenue for Promoting Learning in Computer Science	Peteranetz, Flanigan, Shell, and Soh	2017	IEEE Transactions on Education
[43]	Targeting Metacognition by Incorporating Student-Reported Confidence Estimates on Self-Assessment Quizzes	Lee and Liao	2021	SIGCSE
[85]	The Impact of Programming Project Milestones on Procrastination, Project Outcomes, and Course Outcomes: A Quasi-Experimental Study in a Third-Year Data Structures Course	Shaffer and Kazerouni	2021	SIGCSE
[13]	A Closer Look at Metacognitive Scaffolding: Solving Test Cases Before Programming	Denny, Prather, Becker, Albrecht, Loksa, and Pettit	2019	Koli Calling
[14]	Promoting Early Engagement with Programming Assignments Using Scheduled Automated Feedback	Denny, Whalley, Leinonen	2021	ACE
[44]	Does the Early Bird Catch the Worm? Earliness of Students' Work and its Relationship with Course Outcomes	Leinonen, Castro, Hellas	2021	ITiCSE
[19]	Examining Classroom Interventions to Reduce Procrastination	Martin, Edwards, and Shaffer	2015	ITiCSE
[12]	Self-Predicted and Actual Performance in an Introductory Programming Course	Denny, Luxton-Reilly, Hamer, Dahlstrom, and Purchase	2010	ITiCSE
[46]	Using and Collecting Fine-Grained Usage Data to Improve Online Learning Materials	Leppanen, Leinonen, Ihantola, and Hellas	2017	ICSE
[77]	First Things First: Providing Metacognitive Scaffolding for Interpreting Problem Prompts	Prather, Pettit, Becker, Denny, Loksa, Peters, Albrecht, and Masci	2019	SIGCSE
[9]	Understanding the Relationships Between Self-Regulated Learning and Students Source Code in a Computer Programming Course	Castellanos, Restrepo-Calle, González, and Echeverry	2017	FIE
[2]	In Situ Identification of Student Self-Regulated Learning Struggles in Programming Assignments	Arakawa, Hao, Greer, Ding, Hundhausen, and Peterson	2021	SIGCSE
[41]	Supporting Self-Regulated Learning with Visualizations in Online Learning Environments	Ilves, Leinonen, and Hellas	2018	SIGCSE
[76]	Getting by with help from my friends: Group Study in Introductory Programming Understood as Socially Shared Regulation	Prather, Margulieux, Whalley, Denny, Reeves, Becker, Singh, Powell, and Bosch	2022	ICER
[45]	Pauses and Spacing in Learning to Program	Leppänen, Leinonen, and Hellas	2016	Koli Calling
[86]	A Qualitative Study on How Students Interact with Quizzes and Estimate Confidence on Their Answers	Shah, Lee, Barretto, and Liao	2021	ITiCSE
[8]	Factors for Success in Online CS1	Campbell. Horton, and Craig	2016	ITiCSE
[50]	The Role of Self-Regulation in Programming Problem Solving Process and Success	Loksa and Ko	2016	ICER
[49]	Learning to Program: Gender Differences and Interactive Effects of Students' Motivation, Goals, and Self-Efficacy on Performance	Lishinski, Yadav, Good, and Enbody	2016	ICER
[82]	What Help Do Students Seek in TA Office Hours?	Ren, Krishnamurthi, and Fisler	2019	ICER
[90]	Applying Computational Analysis of Novice Learners' Computer Programming Patterns to Reveal Self-Regulated Learning, Computational Thinking, a nd Learning Performance	Song, Hong, and Oh	2021	Computers in Human Behavior
[96]	Do Students Know What They Think They Know? Assessing Student Confidence in a Computer Graphics Course	Urness	2016	Journal of Computing Sciences in Colleges
[10]	Investigating Patterns of Study Persistence on Self-Assessment Platform of Programming Problem-Solving	Chung and Hsiao	2020	SIGCSE
[37]	Comparison of Grade Replacement and Weighted Averages for Second-Chance Exams	Herman, Cai, Bretl, Zilles, and West	2020	ICER
[78]	Metacognitive Difficulties Faced by Novice Programmers in Automated Assessment Tools	Prather, Pettit, McMurry, Peters, Homer, and Cohen	2018	ICER
[104]	Exploring the Impact of Voluntary Practice and Procrastination in an Introductory Programming Course	Zhang, Cunningham, Iyer, Baker, and Fouh	2022	SIGCSE
[65]	Empirical Evidence for the Existence and Uses of Metacognition in Computer Science Problem Solving	Parham, Gugerty, and Stevenson	2010	SIGCSE
[67]	Struggling To Keep Tabs on Capstone Projects: A Chatbot to Tackle Student Procrastination	Pereira and Díaz	2021	ACM Transactions on Computing Education
[22]	Identifying Computer Science Self-Regulated Learning Strategies	Falkner, Vivian, and Falkner	2014	ITiCSE
[88]	What Do They Note? An Exploratory Investigation Into The Characteristics of CS Students' Notes	Singh, Tempero, Luxton-Reilly, and Zhang	2021	CSERC

Table 8. Papers Reviewed for this Work: Part 1 of 2

Citation	Title	Authors	Year	Published in
[38]	Proof by Incomplete Enumeration and Other Logical Misconceptions	Herman, Kaczmarczyk, Loui, and Zilles	2008	ICER
[94]	Expression of Abstraction: Self explanation in Code Production	Sudol-DeLyser	2015	SIGCSE
[24]	Relationship Between Implicit Intelligence Beliefs and Maladaptive Self-Regulation of Learning	Flanigan, Peteranetz, Shell, and Soh	2023	ACM Transactions on Computing Education
[33]	Self-reported and digital-trace measures of computer science students' self-regulated learning in blended course designs	Han and Ellis	2023	Journal of Education and Information Technologies
[11]	Unobtrusive measurement of self-regulated learning: A clickstream-based multi-dimensional scale	Cristea, Snijders, Matzat and Kleingeld	2023	Journal of Education and Information Technologies
[62]	Investigating the effect of prompts on learners' academic help-seeking behaviours on the basis of learning analytics	Önder and Akçapınar	2023	Journal of Education and Information Technologies
[79]	"It's Weird That it Knows What I Want": Usability and Interactions with Copilot for Novice Programmers	Prather, Reeves, Denny, Becker, Leinonen, Luxton-Reilly, Powell, Finnie-Ansley, and Santos	2023	ACM Transactions on Computing Education
[103]	Assessment of Self-Identified Learning Struggles in CS2 Programming Assignments	Zahn, Gransbury, Heckman and Battestilli	2023	ITICSE
[29]	Supporting Collaboration in Introductory Programming Classes Taught in Hybrid Mode: A Participatory Design Study	Goswami, Zeinoddin, Pegah, Estier and Cherubini	2023	SIGCSE
[1]	Exploring the Differences in Students' Behavioral Engagement With Quizzes and Its Impact on their Performance in a Flipped CS1 Course	Aggarwal and Ashok	2022	Koli Calling
[47]	Examples of Unsuccessful Use of Code Comprehension Strategies: A Resource for Developing Code Comprehension Pedagogy	Lewis	2023	ICER
[89]	Analysis of Novices' Web-Based Help-Seeking Behavior While Programming	Skripchuk, Bennett, Zhang, Li, and Price	2023	SIGCSE
[105]	The Effects of Socially Shared Regulation of Learning on the Computational Thinking, Motivation, and Engagement in Collaborative Learning by Teaching	Zhou and Tsai	2022	Journal of Education and Information Technologies
[102]	Behavioral Consequences of Reminder Emails on Students' Academic Performance: a Real-world Deployment	Ye, Chen, Mao, Wang-Lin, Shaikh, Bernuy, and Williams	2022	SIGITE
[40]	Investigating Reflection in Undergraduate Software Development Teams: An Analysis of Online Chat Transcripts	Hundhausen, Conrad, Adesope, Tariq, Sbai, and Lu	2023	SIGCSE

Table 9. Papers Reviewed for this Work: Part 2 of 2

Key for Conference Acronyms in Table **??**:

- **SIGCSE** - Special Interest Group Computer Science Education Conference
- **FIE** - Frontiers in Education Conference
- **ACE** - Australasian Computing Education Conference
- **ITiCSE** - Innovation and Technology in Computer Science Education Conference
- **ICSE** - International Conference on Software Engineering
- **ICER** - International Computing Education Research Conference
- **CSERC** - Computer Science Education Research Conference
- **SIGITE** - Special Interest Group on Information Technology Education

B Theory-Generated Codes and Final Codes Created

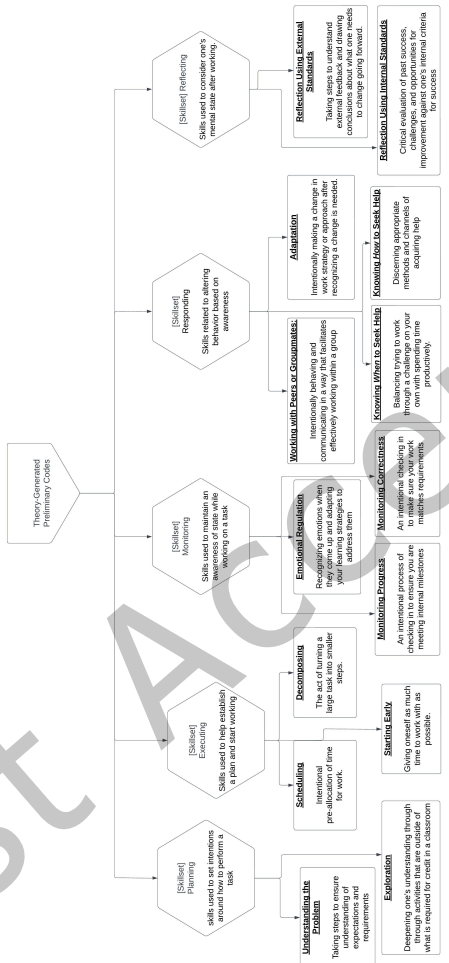

Fig. 4. Diagram showing relationships of theory-based codes from [17]

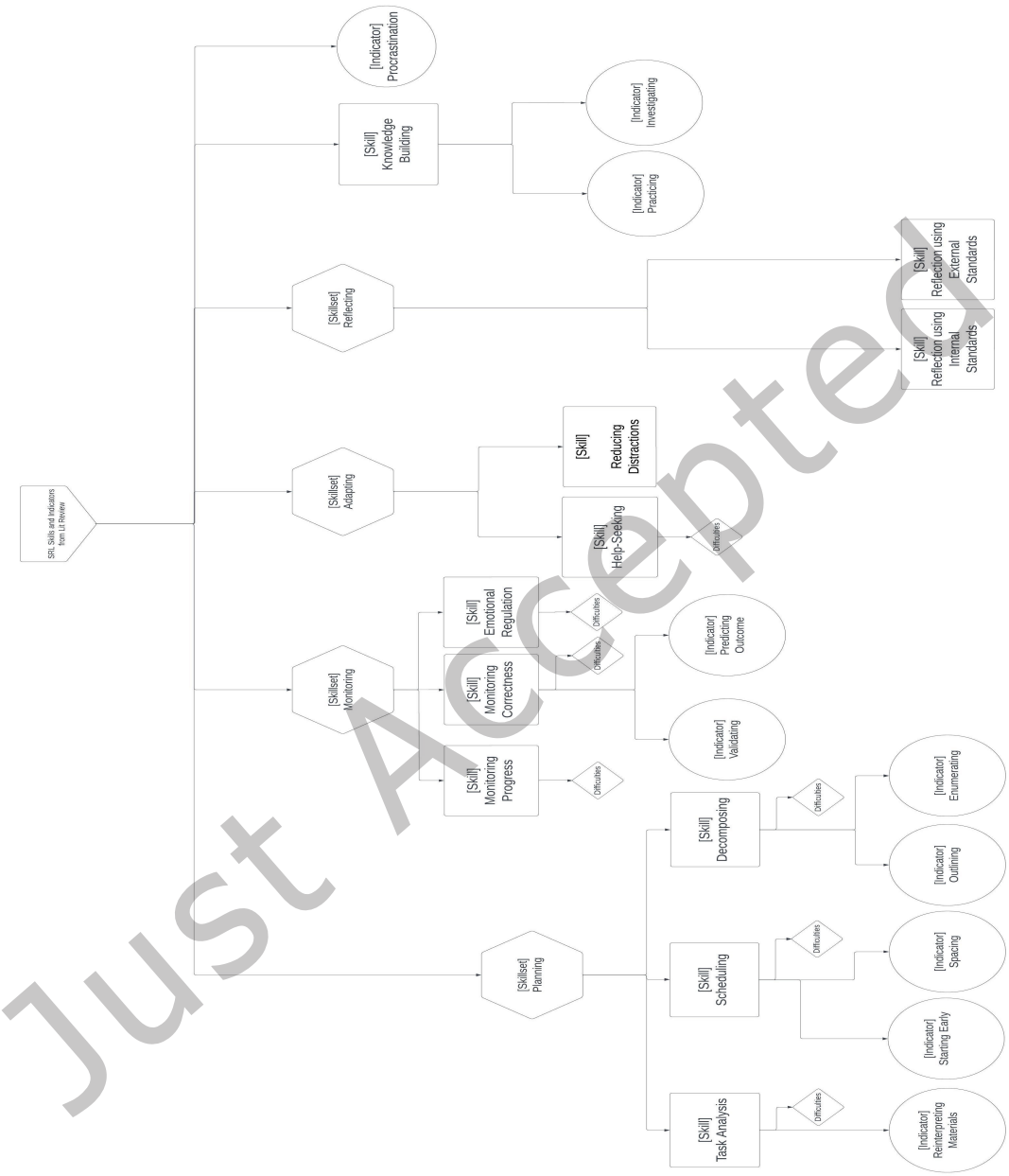

Fig. 5. Final organization for all codes created in this systematic review. Skillsets are hexagons, skills are squares, difficulties are diamonds, and behaviors are circles.

References

[1] Ashish Aggarwal and Akshay Ashok. 2022. Exploring the Differences in Students' Behavioral Engagement With Quizzes and Its Impact on their Performance in a Flipped CS1 Course. In *Proceedings of the 22nd Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '22)*. Association for Computing Machinery, New York, NY, USA, Article 24. https://doi.org/10.1145/3564721.3564740

[2] Kai Arakawa, Qiang Hao, Tyler Greer, Lu Ding, Christopher D. Hundhausen, and Abigayle Peterson. 2021. In Situ Identification Of Student Self-Regulated Learning Struggles In Programming Assignments. 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, 467–473. https://doi.org/10.1145/3408877.3432357

[3] Linda Baker and Ann L Brown. 1980. Metacognitive Skills And Reading. Technical Report No. 188. (1980).

[4] Monique Boekaerts. 2007. Understanding Students' Affective Processes In The Classroom. In *Emotion in Education*. Elsevier, 37–56. https://doi.org/10.1016/B978-012372545-5/50004-6

[5] Monique Boekaerts and Eduardo Cascallar. 2006. How Far Have We Moved Toward The Integration Of Theory And Practice In Self-Regulation? *Educational Psychology Review* 18, 3 (2006), 199–210. https://doi.org/10.1007/s10648-006-9013-4

[6] Jaclyn Broadbent and Walter L Poon. 2015. Self-Regulated Learning Strategies & Academic Achievement In Online Higher Education Learning Environments: A Systematic Review. *The internet and higher education* 27 (2015), 1–13.

[7] Zack Butler, Ivona Bezakova, and Kimberly Fluet. 2017. Pencil Puzzles For Introductory Computer Science: An Experience- And Gender-Neutral Context. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 93–98. https://doi.org/10.1145/3017680.3017765

[8] Jennifer Campbell, Diane Horton, and Michelle Craig. 2016. Factors For Success In Online Cs1. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education* (Arequipa, Peru) *(ITiCSE '16)*. Association for Computing Machinery, New York, NY, USA, 320–325. https://doi.org/10.1145/2899415.2899457

[9] Hugo Castellanos, Felipe Restrepo-Calle, Fabio A. González, and Jhon Jairo Ramírez Echeverry. 2017. Understanding The Relationships Between Self-Regulated Learning And Students Source Code In A Computer Programming Course. In *2017 IEEE Frontiers in Education Conference (FIE)*. 1–9. https://doi.org/10.1109/FIE.2017.8190467

[10] Cheng-Yu Chung and I-Han Hsiao. 2020. Investigating Patterns Of Study Persistence On Self-Assessment Platform Of Programming Problem-Solving. 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, 162–168. https://doi.org/10.1145/3328778.3366827

[11] Tudor Cristea, Chris Snijders, Uwe Matzat, and Ad Kleingeld. 2023. Unobtrusive measurement of self-regulated learning: A clickstream-based multi-dimensional scale. *Education and Information Technologies* 29, 11 (Dec. 2023), 13465–13494. https://doi.org/10.1007/s10639-023-12372-6

[12] Paul Denny, Andrew Luxton-Reilly, John Hamer, Dana B. Dahlstrom, and Helen C. Purchase. 2010. Self-Predicted And Actual Performance In An Introductory Programming Course. In *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education* (Bilkent, Ankara, Turkey) *(ITiCSE '10)*. Association for Computing Machinery, New York, NY, USA, 118–122. https://doi.org/10.1145/1822090.1822124

[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 *Proceedings of the 23rd Australasian Computing Education Conference* (Virtual, SA, Australia) *(ACE '21)*. Association for Computing Machinery, New York, NY, USA, 88–95. https://doi.org/10.1145/3441636.3442309

[15] Flavia Devonas Hoffmann and Kristine Høeg Karlsen. 2022. Choreographic Infrastructuring For Design Things: A New Method For Participatory Design In Teacher Education. (2022), 230–240. https://doi.org/10.1145/3536169.3537796

[16] Augie Doebling and Ayaan M. Kazerouni. 2021. Patterns Of Academic Help-Seeking In Undergraduate Computing Students. In *Proceedings of the 21st Koli Calling International Conference on Computing Education Research* (Joensuu, Finland) *(Koli Calling '21)*. Association for Computing Machinery, New York, NY, USA, Article 13, 10 pages. https://doi.org/10.1145/3488042.3488052

[17] Molly Domino, Bob Edmison, Stephen H Edwards, Rifat Sabbir Mansur, Alexandra Thompson, and Clifford A Shaffer. 2025. Identifying critical self-regulated learning skills: a Delphi process study. *Computer Science Education* 35, 3 (2025), 550–577.

[18] Shannon Duvall, Scott Spurlock, Dugald Ralph Hutchings, and Robert C. Duvall. 2021. Improving Content Learning And Student Perceptions In Cs1 With Scrumage. 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, 474–480. https://doi.org/10.1145/3408877.3432415

[19] Stephen H. Edwards, Joshua Martin, and Clfford A. Shaffer. 2015. Examining Classroom Interventions To Reduce Procrastination. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education* (Vilnius, Lithuania) *(ITiCSE '15)*.

Association for Computing Machinery, New York, NY, USA, 254–259. https://doi.org/10.1145/2729094.2742632

[20] Anastasia Efklides. 2011. Interactions Of Metacognition With Motivation And Affect In Self-Regulated Learning: The Masrl Model. *Educational psychologist* 46, 1 (2011), 6–25. https://doi.org/10.1080/00461520.2011.538645

[21] Anastasia Efklides. 2011. Interactions Of Metacognition With Motivation And Affect In Self-Regulated Learning: The Masrl Model. *Educational psychologist* 46, 1 (2011), 6–25. https://doi.org/10.1080/00461520.2011.538645

[22] 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 & Technology in Computer Science Education* (Uppsala, Sweden) *(ITiCSE '14)*. Association for Computing Machinery, New York, NY, USA, 291–296. https://doi.org/10.1145/2591708.2591715

[23] Abraham E. Flanigan, Markeya S. Peteranetz, Duane F. Shell, and Leen-Kiat Soh. 2015. Exploring Changes In Computer Science Students' Implicit Theories Of Intelligence Across The Semester. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research* (Omaha, Nebraska, USA) *(ICER '15)*. Association for Computing Machinery, New York, NY, USA, 161–168. https://doi.org/10.1145/2787622.2787722

[24] Abraham E. Flanigan, Markeya S. Peteranetz, Duane F. Shell, and Leen-Kiat Soh. 2023. Relationship Between Implicit Intelligence Beliefs and Maladaptive Self-Regulation of Learning. *ACM Trans. Comput. Educ.* 23, 3, Article 32 (June 2023). https://doi.org/10.1145/3595187

[25] John H Flavell. 1979. Metacognition And Cognitive Monitoring: A New Area Of Cognitive–Developmental Inquiry. *American Psychologist* 34, 10 (1979), 906. https://doi.org/10.1037/0003-066X.34.10.906

[26] Carlton J Fong, Cassandra Gonzales, Christie Hill-Troglin Cox, and Holly B Shinn. 2023. Academic Help-Seeking And Achievement Of Postsecondary Students: A Meta-Analytic Investigation. *Journal of Educational Psychology* 115, 1 (2023), 1.

[27] Association for Computing Machinery. 2023. About The Acm Digital Library.

[28] Rita Garcia, Katrina Falkner, and Rebecca Vivian. 2018. Systematic Literature Review: Self-Regulated Learning Strategies Using E-Learning Tools For Computer Science. *Computers & Education* 123 (2018), 150–163. https://doi.org/10.1016/j.compedu.2018.05.006

[29] Lahari Goswami, Pegah Sadat Zeinoddin, Thibault Estier, and Mauro Cherubini. 2023. Supporting Collaboration in Introductory Programming Classes Taught in Hybrid Mode: A Participatory Design Study. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference* (Pittsburgh, PA, USA) *(DIS '23)*. Association for Computing Machinery, New York, NY, USA, 1248–1262. https://doi.org/10.1145/3563657.3596042

[30] Jeffrey Alan Greene and Roger Azevedo. 2007. A Theoretical Review Of Winne And Hadwin'S Model Of Self-Regulated Learning: New Perspectives And Directions. *Review of educational research* 77, 3 (2007), 334–372.

[31] Jeffrey A Greene, Leigh Anna Hutchison, Lara-Jeane Costa, and Helen Crompton. 2012. Investigating How College Students' Task Definitions And Plans Relate To Self-Regulated Learning Processing And Understanding Of A Complex Science Topic. *Contemporary Educational Psychology* 37, 4 (2012), 307–320. https://doi.org/10.1016/j.cedpsych.2012.02.002

[32] Allyson Hadwin, Sanna Järvelä, and Mariel Miller. 2017. Self-Regulation, Co-Regulation, And Shared Regulation In Collaborative Learning Environments. In *Handbook of Self-Regulation of Learning and Performance*. Routledge, 83–106.

[33] Feifei Han and Robert A. Ellis. 2023. Self-reported and digital-trace measures of computer science students' self-regulated learning in blended course designs. *Education and Information Technologies* 28, 10 (March 2023), 13253–13268. https://doi.org/10.1007/s10639-023-11698-5

[34] Brian Harrington, Shichong Peng, Xiaomeng Jin, and Minhaz Khan. 2018. Gender, Confidence, And Mark Prediction In Cs Examinations. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (Larnaca, Cyprus) *(ITiCSE 2018)*. Association for Computing Machinery, New York, NY, USA, 230–235. https://doi.org/10.1145/3197091.3197116

[35] Julian T Hart. 1965. Memory And The Feeling-Of-Knowing Experience. *Journal of educational psychology* 56, 4 (1965), 208.

[36] Hope J Hartman. 2013. *Metacognition in learning and instruction: Theory, research and practice*. Vol. 19. Springer Science & Business Media.

[37] Geoffrey L. Herman, Zhouxiang Cai, Timothy Bretl, Craig Zilles, and Matthew West. 2020. Comparison Of Grade Replacement And Weighted Averages For Second-Chance Exams. 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, 56–66. https://doi.org/10.1145/3372782.3406260

[38] Geoffrey L. Herman, Lisa Kaczmarczyk, Michael C. Loui, and Craig Zilles. 2008. Proof By Incomplete Enumeration And Other Logical Misconceptions. In *Proceedings of the Fourth International Workshop on Computing Education Research* (Sydney, Australia) *(ICER '08)*. Association for Computing Machinery, New York, NY, USA, 59–70. https://doi.org/10.1145/1404520.1404527

[39] Chun-Heng Ho. 2001. Some Phenomena Of Problem Decomposition Strategy For Design Thinking: Differences Between Novices And Experts. *Design Studies* 22, 1 (2001), 27–45.

[40] Christopher Hundhausen, Phill Conrad, Olusola Adesope, Ahsun Tariq, Samir Sbai, and Andrew Lu. 2023. Investigating Reflection in Undergraduate Software Development Teams: An Analysis of Online Chat Transcripts. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 743–749. https://doi.org/10.1145/3545945.3569790

[41] 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

[42] Klaus Krippendorff. 2004. Measuring the reliability of qualitative text analysis data. *Quality and quantity* 38, 6 (2004), 787–800.

[43] Priscilla Lee and Soohyun Nam Liao. 2021. Targeting Metacognition By Incorporating Student-Reported Confidence Estimates On Self-Assessment Quizzes. 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, 431–437. https://doi.org/10.1145/3408877.3432377

[44] Juho Leinonen, Francisco Enrique Vicente Castro, and Arto Hellas. 2021. Does The Early Bird Catch The Worm? Earliness Of Students' Work And Its Relationship With Course Outcomes. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1* (Virtual Event, Germany) *(ITiCSE '21)*. Association for Computing Machinery, New York, NY, USA, 373–379. https://doi.org/10.1145/3430665.3456383

[45] Leo Leppänen, Juho Leinonen, and Arto Hellas. 2016. Pauses And Spacing In Learning To Program. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '16)*. Association for Computing Machinery, New York, NY, USA, 41–50. https://doi.org/10.1145/2999541.2999549

[46] Leo Leppanen, Juho Leinonen, Petri Ihantola, and Arto Hellas. 2017. Using And Collecting Fine-Grained Usage Data To Improve Online Learning Materials. In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*. 4–12. https://doi.org/10.1109/ICSE-SEET.2017.12

[47] Colleen M. Lewis. 2023. Examples of Unsuccessful Use of Code Comprehension Strategies: A Resource for Developing Code Comprehension Pedagogy. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1* (Chicago, IL, USA) *(ICER '23)*. Association for Computing Machinery, New York, NY, USA, 15–28. https://doi.org/10.1145/3568813.3600116

[48] Kathryn Ley and Dawn B Young. 2001. Instructional Principles For Self-Regulation. *Educational Technology Research and Development* 49, 2 (2001), 93–103.

[49] Alex Lishinski, Aman Yadav, Jon Good, and Richard Enbody. 2016. Learning To Program: Gender Differences And Interactive Effects Of Students' Motivation, Goals, And Self-Efficacy On Performance. 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, 211–220. https://doi.org/10.1145/2960310.2960329

[50] 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

[51] 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

[52] Dastyni Loksa, Lauren Margulieux, Brett A. Becker, Michelle Craig, Paul Denny, Raymond Pettit, and James Prather. 2022. Metacognition And Self-Regulation In Programming Education: Theories And Exemplars Of Use. *ACM Trans. Comput. Educ.* 22, 4, Article 39 (sep 2022), 31 pages. https://doi.org/10.1145/3487050

[53] Dastyni Loksa, Benjamin Xie, Harrison Kwik, and Amy J. Ko. 2020. *Investigating Novices' In Situ Reflections On Their Programming Process*. Association for Computing Machinery, New York, NY, USA, 149–155. https://doi.org/10.1145/3328778.3366846

[54] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A. Becker, Michail Giannakos, Amruth N. Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and Claudia Szabo. 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)*. Association for Computing Machinery, New York, NY, USA, 55–106. https://doi.org/10.1145/3293881.3295779

[55] Joseph A. Lyon, Aparajita Jaiswal, Alejandra J. Magana, Ellen Gundlach, and Mark Daniel Ward. 2021. Student Challenges, Strategies, And Learning Within The Data Mine Learning Community. In *2021 IEEE Frontiers in Education Conference (FIE)*. 1–7. https://doi.org/10.1109/FIE49875.2021.9637437

[56] Therese H Macan, Comila Shahani, Robert L Dipboye, and Amanda P Phillips. 1990. College Students' Time Management: Correlations With Academic Performance And Stress. *Journal of educational psychology* 82, 4 (1990), 760.

[57] Jonna Malmberg, Eetu Haataja, and Sanna Järvelä. 2022. Exploring The Connection Between Task Difficulty, Task Perceptions, Physiological Arousal And Learning Outcomes In Collaborative Learning Situations. *Metacognition and Learning* 17, 3 (2022), 793–811. https://doi.org/10.1007/s11409-022-09320-z

[58] Catherine Marshall and Gretchen B Rossman. 2014. Managing, Analyzing, And Interpreting Data. In *Designing Qualitative Research*. Sage Publications, 217–225.

[59] Samiha Marwan, Anay Dombe, and Thomas W. Price. 2020. Unproductive Help-Seeking In Programming: What It Is And How To Address It. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* (Trondheim, Norway) *(ITiCSE '20)*. Association for Computing Machinery, New York, NY, USA, 54–60. https://doi.org/10.1145/3341525.3387394

[60] Giacomo Marzi, Marco Balzano, and Davide Marchiori. 2024. K-Alpha calculator–krippendorff's alpha calculator: a user-friendly tool for computing krippendorff's alpha inter-rater reliability coefficient. *MethodsX* 12 (2024), 102545.

[61] Katherine G. Nelson, Duane F. Shell, Jenefer Husman, Evan J. Fishman, and Leen Kiat Soh. 2015. Motivational And Self-Regulated Learning Profiles Of Students Taking A Foundational Engineering Course. *Engineering education* 104, 1 (1 jan 2015), 74–100. https://doi.org/10.1002/jee.20066 Publisher Copyright: © 2015 ASEE..

[62] Asuman Önder and Gökhan Akçapınar. 2023. Investigating the effect of prompts on learners' academic help-seeking behaviours on the basis of learning analytics. *Education and Information Technologies* 28, 12 (May 2023), 16909–16934. https://doi.org/10.1007/s10639-023-11872-9

[63] Ernesto Panadero. 2017. A Review Of Self-Regulated Learning: Six Models And Four Directions For Research. *Frontiers in psychology* (2017), 422. https://doi.org/10.3389/fpsyg.2017.00422

[64] Ernesto Panadero, Julia Klug, and Sanna Järvelä. 2016. Third wave of measurement in the self-regulated learning field: When measurement and intervention come hand in hand. *Scandinavian Journal of Educational Research* 60, 6 (2016), 723–735.

[65] Jennifer Parham, Leo Gugerty, and D. E. Stevenson. 2010. Empirical Evidence For The Existence And Uses Of Metacognition In Computer Science Problem Solving. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education (SIGCSE '10)*. Association for Computing Machinery, New York, NY, USA, 416–420. https://doi.org/10.1145/1734263.1734406

[66] Eric J Paulson and Laurie Bauer. 2011. Goal Setting As An Explicit Element Of Metacognitive Reading And Study Strategies For College Reading. *NADE Digest* 5, 3 (2011), 41–49.

[67] Juanan Pereira and Óscar Díaz. 2021. Struggling To Keep Tabs On Capstone Projects: A Chatbot To Tackle Student Procrastination. *ACM Trans. Comput. Educ.* 22, 1, Article 4 (oct 2021), 22 pages. https://doi.org/10.1145/3469127

[68] Markeya S. Peteranetz, Abraham E. Flanigan, Duane F. Shell, and Leen-Kiat Soh. 2017. Computational Creativity Exercises: An Avenue For Promoting Learning In Computer Science. *IEEE Transactions on Education* 60, 4 (2017), 305–313. https://doi.org/10.1109/TE.2017.2705152

[69] Paul R Pintrich et al. 1991. A Manual For The Use Of The Motivated Strategies For Learning Questionnaire (MSLQ). (1991). https://files.eric.ed.gov/fulltext/ED338122.pdf

[70] 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. https://doi.org/10.1037/0022-0663.82.1.33

[71] Paul R Pintrich, David AF 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

[72] Paul R Pintrich, Christopher A Wolters, and Gail P Baxter. 2000. Assessing Metacognition And Self-Regulated Learning. In *Issues in the Measurement of Metacognition*. 43–97.

[73] Annette Plüddemann, Jeffrey K Aronson, Igho Onakpoya, Carl Heneghan, and Kamal R Mahtani. 2018. Redefining Rapid Reviews: A Flexible Framework For Restricted Systematic Reviews. *BMJ Evidence-Based Medicine* 23, 6 (2018), 201–203. https://doi.org/10.1136/bmjebm-2018-110990 arXiv:https://ebm.bmj.com/content/23/6/201.full.pdf

[74] Pierre Pluye, Quan Nha Hong, and Isabelle Vedel. 2016. Toolkit For Mixed Studies Reviews (V3).

[75] 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

[76] 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 - Volume 1* (Lugano and Virtual Event, Switzerland) *(ICER '22)*. Association for Computing Machinery, New York, NY, USA, 164–176. https://doi.org/10.1145/3501385.3543970

[77] 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

[78] 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

[79] James Prather, Brent N. Reeves, Paul Denny, Brett A. Becker, Juho Leinonen, Andrew Luxton-Reilly, Garrett Powell, James Finnie-Ansley, and Eddie Antonio Santos. 2023. "It's Weird That it Knows What I Want": Usability and Interactions with Copilot for Novice Programmers. *ACM Trans. Comput.-Hum. Interact.* 31, 1, Article 4 (Nov. 2023). https://doi.org/10.1145/3617367

[80] Minna Puustinen and Lea Pulkkinen. 2001. Models Of Self-Regulated Learning: A Review. *Scandinavian journal of educational research* 45, 3 (2001), 269–286.

[81] Jhon Jairo Ramírez Echeverry, Àgueda García Carrillo, and Fredy Andres Olarte Dussan. 2016. Adaptation And Validation Of The Motivated Strategies For Learning Questionnaire-MSLQ-In Engineering Students In Colombia. *International journal of engineering education* 32, 4 (2016), 1774–1787.

[82] Yanyan Ren, Shriram Krishnamurthi, and Kathi Fisler. 2019. What Help Do Students Seek In Ta Office Hours?. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19)*. Association for Computing Machinery, New York, NY, USA, 41–49. https://doi.org/10.1145/3291279.3339418

[83] Anne Roth, Sabine Ogrin, and Bernhard Schmitz. 2016. Assessing Self-Regulated Learning In Higher Education: A Systematic Literature Review Of Self-Report Instruments. *Educational Assessment, Evaluation and Accountability* 28, 3 (2016), 225–250.

[84] Phil Sands and Aman Yadav. 2020. *Self-Regulation For High School Learners In A Mooc Computer Science Course.* Association for Computing Machinery, New York, NY, USA, 845–851. https://doi.org/10.1145/3328778.3366818

[85] Clifford A. Shaffer and Ayaan M. Kazerouni. 2021. The Impact Of Programming Project Milestones On Procrastination, Project Outcomes, And Course Outcomes: A Quasi-Experimental Study In A Third-Year Data Structures Course. In *Proceedings of the 52nd ACM Technical Symposium on Computer AScience Education* (Virtual Event, USA) *(SIGCSE '21)*. Association for Computing Machinery, New York, NY, USA, 907–913. https://doi.org/10.1145/3408877.3432356

[86] Kartik Shah, Priscilla Lee, Daphne Barretto, and Soohyun Nam Liao. 2021. A Qualitative Study On How Students Interact With Quizzes And Estimate Confidence On Their Answers. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1* (Virtual Event, Germany) *(ITiCSE '21)*. Association for Computing Machinery, New York, NY, USA, 32–38. https://doi.org/10.1145/3430665.3456377

[87] 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

[88] Paramvir Singh, Ewan Tempero, Andrew Luxton-Reilly, and Shuxiang Zhang. 2022. What Do They Note? An Exploratory Investigation Into The Characteristics Of Cs Students' Notes. In *Proceedings of the 10th Computer Science Education Research Conference* (Virtual Event, Netherlands) *(CSERC '21)*. Association for Computing Machinery, New York, NY, USA, 57–67. https://doi.org/10.1145/3507923.3507934

[89] James Skripchuk, Neil Bennett, Jeffrey Zhang, Eric Li, and Thomas Price. 2023. Analysis of Novices' Web-Based Help-Seeking Behavior While Programming. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1* (Toronto ON, Canada) *(SIGCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 945–951. https://doi.org/10.1145/3545945.3569852

[90] Donggil Song, Hyeonmi Hong, and Eun Young Oh. 2021. Applying Computational Analysis of Novice Learners' Computer Programming Patterns to Reveal Self-Regulated Learning, Computational Thinking, and Learning Performance. *Computers in Human Behavior* 120 (2021), 106746. https://doi.org/10.1016/j.chb.2021.106746

[91] Kristin Stephens-Martinez. 2021. A Study Of The Relationship Between A Cs1 Student'S Gender And Performance Versus Gauging Understanding And Study Tactics. 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, 679–685. https://doi.org/10.1145/3408877.3432365

[92] Ben Stephenson, Michelle Craig, Daniel Zingaro, Diane Horton, Danny Heap, and Elaine Huynh. 2017. Exam Wrappers: Not A Silver Bullet. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 573–578. https://doi.org/10.1145/3017680.3017701

[93] Robert J Sternberg et al. 1985. *Beyond Iq: A Triarchic Theory Of Human Intelligence.* CUP Archive.

[94] Leigh Ann Sudol-DeLyser. 2015. Expression Of Abstraction: Self Explanation In Code Production. 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, 272–277. https://doi.org/10.1145/2676723.2677222

[95] Lauren C. Hensley Sungjun Won and Christopher A. Wolters. 2021. Brief Research Report: Sense of Belonging and Academic Help-Seeking as Self-Regulated Learning. *The Journal of Experimental Education* 89, 1 (2021), 112–124. https://doi.org/10.1080/00220973.2019.1703095 arXiv:https://doi.org/10.1080/00220973.2019.1703095

[96] Timothy Urness. 2016. Do Students Know What They Think They Know? Assessing Student Confidence In A Computer Graphics Course. *J. Comput. Sci. Coll.* 31, 5 (May 2016), 140–146. https://doi.org/10.5555/2904298.2904325

[97] Philip H Winne. 2011. A Cognitive And Metacognitive Analysis Of Self-Regulated Learning. *Handbook of self-regulation of learning and performance* (2011), 15–32.

[98] Philip H Winne. 2013. Self-Regulated Learning Viewed From Models Of Information Processing. In *Self-Regulated Learning and Academic Achievement*. Routledge, 145–178.

[99] Philip H Winne. 2017. Learning Analytics For Self-Regulated Learning. *Handbook of Learning Analytics* 754 (2017), 241–249. https://doi.org/10.18608/hla22.008

[100] Philip H Winne. 2017. Learning analytics for self-regulated learning. *Handbook of learning analytics* 754 (2017), 241–249.

[101] Philip H Winne and Nancy E Perry. 2006. Learning from learning kits: gStudy traces of students' self-regulated engagements with computerized content. *Educational Psychology Review* 18, 3 (2006), 211–228.

[102] Runlong Ye, Pan Chen, Yini Mao, Angela Wang-Lin, Hammad Shaikh, Angela Zavaleta Bernuy, and Joseph Jay Williams. 2022. Behavioral Consequences of Reminder Emails on Students' Academic Performance: a Real-world Deployment. In *Proceedings of the 23rd Annual Conference on Information Technology Education* (Chicago, IL, USA) *(SIGITE '22)*. Association for Computing Machinery, New York, NY, USA, 16–22. https://doi.org/10.1145/3537674.3554740

[103] Matthew Zahn, Isabella Gransbury, Sarah Heckman, and Lina Battestilli. 2023. Assessment of Self-Identified Learning Struggles in CS2 Programming Assignments. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1* (Turku, Finland) *(ITiCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 264–270. https://doi.org/10.1145/3587102.3588786

[104] Jiayi Zhang, Taylor Cunningham, Rashmi Iyer, Ryan Baker, and Eric Fouh. 2022. Exploring The Impact Of Voluntary Practice And Procrastination In An Introductory Programming Course. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1* (Providence, RI, USA) *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 356–361. https://doi.org/10.1145/3478431.3499350

[105] Xiaodan Zhou and Chia-Wen Tsai. 2022. The Effects of Socially Shared Regulation of Learning on the Computational Thinking, Motivation, and Engagement in Collaborative Learning by Teaching. *Education and Information Technologies* 28, 7 (Dec. 2022), 8135–8152. https://doi.org/10.1007/s10639-022-11527-1

[106] Barry J Zimmerman and Adam R Moylan. 2009. Self-Regulation: Where Metacognition And Motivation Intersect. In *Handbook of metacognition in education*. 311–328. https://doi.org/10.4324/9780203876428.ch16

[107] 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