Computational Thinking for Self-Regulated Learning

Stefan Pasterk*
Gertraud Benke*
stefan.pasterk@aau.at
gertraud.benke@aau.at
University of Klagenfurt
Klagenfurt, Austria

ABSTRACT

In this theoretical paper, we compare computational thinking and self-regulated learning. Many studies use self-regulated learning to foster the acquisition of computational thinking competencies. Self-regulated learning skills are themselves beneficial for any learning process; here, we argue that the relationship between self-regulated learning and computational thinking is closer than the simple observation that self-regulated learning strategies support the acquisition of computational thinking competencies. We sustain that self-regulated learning and computational thinking competencies share many features (and have some differences), which would support synergistic effects so that not only can self-regulated learning be used to attain computational thinking competencies, but computational thinking activities can also be used to foster features of self-regulated learning competencies.

CCS CONCEPTS

• **Social and professional topics → Computational thinking**; **K-12 education**; *Informal education.*

KEYWORDS

self-regulated learning, computational thinking, secondary school

ACM Reference Format:
Stefan Pasterk and Gertraud Benke. 2024. Computational Thinking for Self-Regulated Learning. In *Proceedings of the 2024 Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024), July 8–10, 2024, Milan, Italy.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3649217.3653565

1 INTRODUCTION

In the last decade, studies explored how self-regulated learning can be used to support the learning of computational thinking (e.g. [6], [35]). In this paper, we argue that these concepts share many aspects. Therefore, not only may self-regulated learning competencies facilitate computational thinking, but the acquisition of computational thinking may also facilitate the acquisition of self-regulated learning competencies. As computational thinking is part of the national curriculum in some countries (e.g., Austria), whereas self-regulated learning has not been included in national curricula (as far as we are aware), teaching computational thinking may also be a means to foster cross-curricular competencies in self-regulated learning. In this short paper, we will discuss both concepts, elaborating on their similarities and differences.

We will first briefly present features of computational thinking and self-regulated learning, and discuss studies on self-regulated learning and computational thinking. Next, we will describe the similarity of particular features of both concepts, and point out their differences. Finally, we argue for deliberately also using computational thinking activities as means for the acquisition of self-regulated learning competencies.

2 COMPUTATIONAL THINKING

Computational Thinking (CT) has been a point of interest in computer science and its education over the last 30 years [2], [9]. With Jeannette Wing's frequently cited article [31], it became known to a more general audience like policymakers. This led to the inclusion of CT in several curricula or educational programs worldwide. With the broader use of the term 'computational thinking', different definitions and interpretations emerged, many of which are presently in use [15], [25]. Denning distinguishes two views of CT [9]. The traditional one is used in computer science; there, CT is seen as a set of skills for the design and implementation of software. A frequently referenced definition comes from Aho [1], who emphasizes the meaning of computational models: *'Mathematical abstractions called models of computation are at the heart of computation and computational thinking. Computation is a process that is defined in terms of an underlying model of computation, and computational thinking is the thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms.'* Compared to that view, in the new view, CT is seen as a framework that enables programming for all people, not only software engineers. It is described as a set of skills or mental tools that everyone can or should acquire to solve problems or develop systems based on concepts from computer science. The basic idea is to turn a difficult problem into a problem we know how to solve using reduction, embedding, transformation, or simulation. For Wing, CT 'represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use' and includes, besides others thinking recursively, parallel processing, abstraction, decomposition, program correctness and efficiency, and error correction. The main characteristics of this view of CT are [31]:

- Conceptualizing, not programming
- Fundamental, not rote skill
- A way that humans, not computers, think

*Both authors contributed equally to this research.

ITiCSE 2024, July 8–10, 2024, Milan, Italy

ACM ISBN 979-8-4007-0600-4/ 24/07
https://doi.org/10.1145/3649217.3653565

- Complements and combines mathematical and engineering thinking
- Ideas, not artifacts
- For everyone, everywhere

Clearly, this view distances itself from software development and focuses on everyday problems. This idea of CT became popular among educators and policymakers, who called for integrating CT into school teaching [14]. Based on this and a later definition from Wing the K-12 Computer Science Framework defines CT as follows [8]: *'Computational thinking refers to the thought processes involved in expressing solutions as computational steps or algorithms that can be carried out by a computer.'* The new view can also be seen as a basic version of CT, or *basic CT*, for beginners [10]. Proponents of CT break CT down into six key competencies *decomposition*, *abstraction*, *algorithms*, *debugging*, *iteration*, and *generalization* [25]. Table 1 shows these facets, including the definition by Shute, Sun, and Clarke [25].

The facets show a general and analytical view of CT. They are not unique to computing but can be found in different disciplines and can, therefore, be helpful in several activities beyond computer science [29]. This view of CT has been strongly promoted for some time now. In recent years, it has been pointed out that teaching CT without its background concepts of computer science can lead to comprehension problems. Denning and Tedre see algorithms, programming, and machines as central for basic and advanced CT, as people have difficulties understanding CT concepts without understanding the 'machine in the background'. Unique computing or computer science skills come with an advanced form of CT. They relate an advanced CT with skills in the following areas [10]:

- designing and building large, reliable, and safe software
- simulations
- artificial intelligence
- performance evaluation of systems
- distributed networks
- operating systems
- interfaces for complex systems

These topics demonstrate the clear connection between CT and computer science. Where basic CT is very general and - with or without the context of computing machines - can be found in different fields or subjects, advanced CT covers everything that goes deeper into the details of computer science.

In this contribution, the more general aspects of CT, so basic CT, are addressed, focusing on the abilities learners can acquire through education in CT. Whether the training occurred in connection with computer science or not is not discussed here.

3 SELF-REGULATED LEARNING

According to Schunk and Greene ([22, p. 1]), 'Self-regulation refers to the ways that learners systematically activate and sustain their cognitions, motivations, behaviors, and affects, toward the attainment of their goals.' Thus, self-regulation, understood as a cognitive mechanism to guide one's behavior, depends on metacognitive competencies. Therefore, self-regulation is usually regarded as a competence (e.g., [11]). In the literature discussed in the next section on the intersection of SRL and CT, the capabilities involved are usually discussed as skills (e.g. [35], [34]). It would exceed the scope of this paper to explore this distinction; in this paper, we will follow the usage of the SRL community. Self-regulated learning occurs when students deliberately engage in goal-directed activities using task-relevant strategies. Self-regulation focuses on the learning processes needed rather than the goal itself. Successful self-regulation often requires a cyclical and dynamic process, including feedback loops. Models of self-regulated learning usually look at the process learners are engaged in when striving to reach a goal or, more concretely, when they want to complete a task. Like models in motivation theory (e.g., expectancy-value theory, [13]), researchers distinguish between a planning, process, and evaluation phase. Self-regulation requires different (meta-)cognitive engagements in each phase; however, students might self-regulate only some of the phases and follow routines and unconscious strategies during other phases. In addition, if the work progresses smoothly, self-regulation may not be necessary for successful task completion. Thus, considering an already defined task, self-regulation is necessary when routines fail or students evaluate their task completion. The planning phase comprises immediate and general consideration about planning the task (as in expectancy-value theory). Is the time right, the situation supportive enough, and the task important enough to do it at all? Should the task be postponed, or has something to be changed in the environment before starting with the activity? In the immediate planning phase, self-regulated students structure the task into sub-tasks with clear and attainable goals, thinking about the process and the strategies they want to pursue to reach their goals. In the process phase, a self-regulated learner will monitor their progress; in particular when pursuing unfamiliar or difficult tasks. There, self-regulation includes self-observation, self-evaluation, and self-reaction [28, p. 22], as students provide feedback on their own doing and change their processes as needed. This phase's key elements are time management, effort regulation, and help-seeking. If the task is a learning task (e.g. students should study a text), self-regulated students will also make use of appropriate learning strategies like organization and elaboration strategies. Finally, after task completion, in the evaluation phase, self-regulated learners might review the process and assess the outcome, providing self-feedback on task engagement and drawing conclusions for future engagements. They may change their overarching goals, review strategies used, learn from the process on the content level, and extract strategies for future purposeful engagement.

The enactment of self-regulated learning depends on students' capacities and the learning environment's affordances. Many learning environments do not allow students to set their own learning goals, which denies them the ability to self-regulate their goal-setting and engage deeply in the first phase of self-regulated learning. Within this general framework, researchers have developed different models of self-regulated learning (see [16]), highlighting different aspects and including additional elements in their consideration of student learning. E.g. Winne's four-stage model ([32]) considers conditions, operations, products, standards, and evaluations. Conditions are what the learner believes impacts their work on a task, as well as all personal factors of the learner (prior knowledge, motivation, emotion, etc.). Operations are information processing operations and more general strategies when working with information. Operations generate products, which are then evaluated using some standard. Boekaerts [4] has mapped

Table 1: Facets of CT [25]

Facet	Definition
Decomposition	Dissect a complex problem/system into manageable parts. The divided parts are not random pieces, but functional elements that collectively comprise the whole system/problem.
Abstraction	Extract the essence of a (complex) system. Abstraction has three subcategories: (a) Data collection and analysis: Collect the most relevant and important information from multiple sources and understand the relationships among multilayered datasets; (b) Pattern recognition: Identify patterns/rules underlying the data/information structure; (c) Modeling: Build models or simulations to represent how a system operates, and/or how a system will function in the future.
Algorithms	Design logical and ordered instructions for rendering a solution to a problem. The instructions can be carried out by a human or computer. There are four sub-categories: (a) Algorithm design: Create a series of ordered steps to solve a problem; (b) Parallelism: Carry out a certain number of steps at the same time; (c) Efficiency: Design the fewest number of steps to solve a problem, removing redundant and unnecessary steps; (d) Automation: Automate the execution of the procedure when required to solve similar problems.
Debugging	Detect and identify errors, and then fix the errors, when a solution does not work as it should.
Iteration	Repeat design processes to refine solutions, until the ideal result is achieved.
Generalization	Transfer CT skills to a wide range of situations/domains to solve problems effectively and efficiently.

out six components that all contribute to self-regulation: cognitive and motivational *regulatory* strategies, cognitive and motivational strategies, and knowledge and beliefs in the content domain as well as self-knowledge and self-belief. In her later dual processing self-regulation model, Boekaerts [5] distinguishes between two distinct 'pathways' of SRL: the growth pathway that is concerned with growth in competencies and describes the use of metacognitive strategies facilitating (cognitive) learning, and the well-being pathway that addresses emotional-motivational aspects. This model highlights the observation that students with high proficiency in self-regulation may choose not to engage in certain tasks [30]. These later approaches to analyzing self-regulated learning focus on the different aspects involved in SRL instead of analyzing the process of engaging in a self-regulated activity. Self-regulated learning, as described above, has to be learned itself. As Wesarg-Mezel and colleagues note [30, p. 1], studies have predominantly focused on SRL abilities, and less on developmental aspects. Consequently, we have good models about the structure of SRL, about which aspects of SRL are supportive in which phase of task engagement, but we know much less on a theoretical level about how to develop SRL. Most of the interventions use some form of teacher-guided or written prompts to direct learners' attention toward desirable SRL activities (planning, evaluation, etc.).

4 SRL TO FOSTER CT

It is well established, and from the perspective of learning theory not surprising that SRL generally supports academic achievement [3]. Many studies look at the correlation of SRL scores or some sub-component (each determined by self-report questionnaires or teacher ratings, see [33]) with other target variables, like academic performance, or use event-sampling to understand more about the processes of SRL [7]. The finding that SRL supports learning, has led to proposals of how to facilitate SRL-strategies in various subject areas (see Section II in [22]). Didactic interventions use prompts to guide students to reflect on their goals, on whether they want to change something in their environment, formulate explicit plans, include reflection and evaluation prompts. These prompts should at the same time facilitate a practice of this kind of reflection, and lead to a habit of self-generating these kinds of questions in similar situations.

For CT, Zhou et al. [35, p. 404] find: 'A review of existing research reveals that there are few empirical studies on the intersection of SDL and CT [..].' Most of these studies have been looking at how SRL or self-directed learning (SDL) (which is - depending on the author - just another name for SRL or a similar construct with more focus on environmental aspects of learning, see [21]) can support CT. There, SRL competencies (or skills for Zhou et al.) are seen as facilitators of CT, enabling students "to initiate and guide their use of CT". When one considers that SRL is the competence to orient oneself to a goal, monitor one's progress, and evaluate the result, it will not surprise that studies find that competencies in SDL or SRL foster CT. Peters-Burton et al. [17, p. 257] conclude that "because computational thinking processes can be viewed as goal-directed processes, it is possible to use self-regulated learning theory as a framework for assessing and enhancing computational thinking". The authors also point out the similarities between CT and SRL: "[..] CPs [computational practices] parallels [sic!] the regulatory process because it involves making a series of judgements toward a

Table 2: SRL and CT

Self-Regulated Learning	Computational Thinking
planning - goal considerations	abstraction decomposition
planning - structuring of the task	decomposition abstraction algorithms
monitor progress and reviews as needed	debugging iteration abstraction
evaluation of result	debugging iteration
evaluation for further progress	generalization

solution to a problem, deploying tactics to conceptualize problems iteratively, identifying patterns in data such as points of divergence and convergence, and troubleshooting problems during the process. Also similar to SRL, to engage in the iterative processes of CT, students must be motivated to do so, possess knowledge of tactics to reach their goals, and effectively monitor and evaluate how well they are proceeding to a problem solution." [17, p. 258/259] Despite these structural similarities, many studies (e.g. [24], [26], [23]) use a SRL-questionnaire to determine a single SRL-score, which is then related to features of CT. A notable exception is [34], who analyzed the correlation of sub-features of pre-service teachers' CT and SRL skills. Zhou et al. [35] is one of the few intervention studies. The authors included teacher scaffolding for SRL (like articulating the learning objective) and self-directed learning prompts for a group of students in a coding course, and found that group to outperform a control group in their CT (without measuring individual SRL competencies).

5 A CONCEPTUAL COMPARISON

If we compare SRL and CT (Table 2), we find that many features of CT are already present in the very first phase of goal-setting planning. Goal setting and planning involves careful consideration of the problem at hand, a first determination on whether one is motivated to pursue the task, formulating a goal and deciding on first steps, and identifying the strategies that will be used to accomplish them: 'Task analysis involves considering what will be required for successful action, breaking down complex tasks into manageable components' [28, p. 24]. With Suchman [27], we do not want to assume that problem-solvers always or frequently form a fully developed plan towards the final goal before they start their engagement (although, at times, they may). Still, we assume a more dynamic and situated approach to problem-solving. This means students will start in a promising direction and only consider the following steps once they have finished the present engagement, whether they have reached their sub-goal or encountered difficulties. Then, they must evaluate their new position and engage in a new planning process. These activities require a certain degree of abstraction in the first step. The problem must be recognized and analyzed to set the goals appropriately. For this purpose, an overview or general perspective of the problem is helpful, if not necessary. This can be achieved with the help of abstraction, as described by Grover and Pea: 'abstraction reduces information and detail to focus on concepts relevant to understanding and solving problems' [15, p. 39]. It is also necessary to have a basic subdivision of a problem into smaller steps to formulate sub-goals. Only if these steps are recognized can sub-goals be specified and organized. This means that the decomposition into smaller problems understood as 'breaking problems down into smaller parts that may be more easily solved' [2, p. 52], is also helpful when approaching a new problem. Therefore, competence with these two facets of CT - abstraction and decomposition - might facilitate students' goal considerations. Students might be able to navigate in a more structured way in learning environments requiring self-regulation.

In the planning process, they necessarily engage in a process of decomposition of the problem; frequently, one of the (sub-)goals they need to address will involve considering particular data and determining relevant and irrelevant information. Depending on the problem, students need to engage in pattern recognition and modeling. In other words, the phase of planning will contain elements of abstraction. In addition to collecting and analyzing data, creating, manipulating, and visualizing data are CT practices [29] that can help prepare students for various tasks at this stage. CT enables them to structure data, processes, or problems using different models to understand concepts or find and test solutions [29]. Depending on their expertise, age, and the problem at hand, students will develop more complex and deliberate plans to address the problem - using an algorithmic design - or engage in situated cognition, in which they mainly strive to solve the problem one step at a time, without a more abstract and holistic consideration of all the steps taken or still needed. Algorithmic thinking is a major facet of CT and includes sequential representation of steps and even more complex concepts. Having prior knowledge of algorithms or programming can have an impact on the methods used by students.

During the execution of the plan, SRL requires a constant process of monitoring the progress and quality of the work and corrections as needed. While more significant problems might necessitate the revision of the entire plan, returning to phase one (starting the next iteration), this phase includes minor corrections and deliberate guidance of one's activity. One example might be our writing of this article, in which we constantly formulate thoughts within the overall plan for the present section, revise our sentences, and correct spelling mistakes while writing. In CT terminology, this can also be understood as iteration and debugging. Iteration is the repetition of any step in the design process as many times as necessary to achieve an ideal result (see table 1) [25]. These can be smaller review loops during the development or refinement of a solution and larger loops for reviewing an entire approach proposal. Review loops aim to implement possible improvements to the solution, including the systematic detection, identification, and correction of errors. As solutions for sub-goals should also be evaluated, there is also a link to abstraction.

At the end of the activity, students engaged in SRL should evaluate their results - this can again be likened to debugging and entering a possible iterative process of change and further checking. As already described, this represents a large review loop where not only smaller parts but the entire solution is evaluated. Finally, students should reflect on the task, what aspects of their approach proved successful, and which steps should be avoided in a similar task. This can be understood as engaging in prospective transfer or generalization. As the definition of generalization according to Shute, Sun, and Clarke describes, this involves the transfer of CT strategies to different contexts [25]. This means that students are aware that they can apply CT skills to other situations and problems and thus master effective tools for solving them.

Overall, we find correspondences to most core processes of engaging in SRL, but some (sub)facets of CT are not reflected: automation (for trivial reasons), parallelism, and efficiency. The first phase of SRL (goal setting and planning) especially contains several CT facets, indicating the complexity of this phase. At the same time, there is a clear focus on the cognitive side or growth pathway of SRL; goal-setting (SRL) does not have an equivalent in CT, and emotional-motivational aspects are also not part of the feature structure of CT.

6 DISCUSSION AND CONCLUSION

We are not the first to point out the relationship between SRL and CT [17]. In their comparison, peters-Burton, Cleary, and Kitsantas conclude, 'SRL theory provides a compelling and integrative framework from which CT skills can be taught.' [17, p. 260]. Their article sees SRL as a learning theory that can support and structure CT instruction. In here, we argue for another relationship between these two concepts. SRL is an important element in a learning theory and something we would like to instill in learners as a habituated practice. However, despite more than 20 years of research on SRL and its impact on academic performance, SRL is generally not part of national curricula despite calls for its inclusion [19]. At the same time, CT has been included in several national and state curricula (see [20]) and is, therefore, a necessary element in every child's education. Given the conceptual similarities of CT and (growth-pathway) SRL, instruction in CT can be seen as supporting certain features of SRL or bluntly, as a special form of applied SRL. In particular, the careful determination of goals and sub-goals, conditions of their attainability, and other aspects considered during planning when engaged in CT are essential aspects for any deliberate reflection on any task. These considerations are prerequisites for later reflections on the task progress and evaluating the task attainment. Instruction in CT, particularly in a formal context with a well-defined problem, may serve as a practice ground to engage the kind of questions SRL also requires. Debugging, conversations about efficiency (usually not included in SRL), or beautiful coding (if the activity involves that) allow a very concrete conversation about quality and processes it might need to raise the quality of some work and thus introduce evaluation as the final part of task completion. Iterative processes, either as part of the problem-solving ask or else as part of the debugging process, offer the possibility to compare this process to other problem-solving activities and to include conversations on a growth mindset [12]. This, in turn, could address the motivational pathway of SRL.

Considering CT as a way to foster SRL requires refocusing some research questions and presentations. While present studies look at the relationship between the two constructs, only a few look 'below' the construct to the relationship between individual components of these constructs. We need a better understanding of which components have a close and which a loose relationship. We need to learn more about the specific implementations along the lines of [18, section 5] to make inferences about which scaffolds were present, which might be lacking, which might be important to both SRL and CT and which are only for one of them. And how can curricular material and instruction in CT be characterized regarding the cognitive and motivational strategies being engaged?

Furthermore, we conjecture that teachers might see CT as more concrete (teaching particular topics as determined in the curriculum and nationally sanctioned school books) and, therefore, easier than teaching for SRL.

REFERENCES

[1] Alfred V. Aho. 2011. Ubiquity Symposium: Computation and Computational Thinking. 2011, January (2011). https://doi.org/10.1145/1922681.1922682

[2] Valerie Barr and Chris Stephenson. 2011. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? 2, 1 (2011). https://doi.org/10.1145/1929887.1929905

[3] Friederike Blume, Andrea Irmer, Judith Dirk, and Florian Schmiedek. 2022. Day-to-day variation in students' academic success: The role of self-regulation, working memory, and achievement goals. *Developmental science* 25, 6 (2022), e13301. https://doi.org/10.1111/desc.13301

[4] Monique Boekaerts. 1996. Self-regulated Learning at the Junction of Cognition and Motivation. *European Psychologist* 1, 2 (1996), 100–112. https://doi.org/10.1027/1016-9040.1.2.100

[5] Monique Boekaerts. 2011. Emotions, emotion regulation, and self-regulation of learning. In *Handbook of self-regulation of learning and performance.* Routledge/Taylor & Francis Group, New York, NY, US, 408–425.

[6] Mic Bowman, Saumya K. Debray, and Larry L. Peterson. 1993. Reasoning About Naming Systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (November 1993), 795–825. https://doi.org/10.1145/161468.161471

[7] Timothy J. Cleary and Callan, Gregory, L. 2018. Assessing Self-Regulated Learning Using Microanalytic Methods. In *Handbook of self-regulation of learning and performance*, Dale H. Schunk and Jeffrey A. Greene (Eds.). Routledge Taylor & Francis Group, New York and London, 338–351.

[8] K-12 Computer Science Framework Steering Committee. 2016. *K-12 Computer Science Framework.* Technical Report. New York, NY, USA.

[9] Peter J. Denning. 2017. Remaining Trouble Spots with Computational Thinking. 60, 6 (2017). https://doi.org/10.1145/2998438

[10] Peter J. Denning and Matti Tedre. 2022. Computational Thinking: A Disciplinary Perspective. *Informatics in Education* 20, 3 (2022), 361–390. https://doi.org/10.15388/infedu.2021.21

[11] Markus Dresel, Bernhard Schmitz, Barbara Schober, Christine Spiel, Albert Ziegler, Tobias Engelschalk, Gregor Jöstl, Julia Klug, Anne Roth, Bastian Wimmer, and Gabriele Steuer. 2015. Competencies for successful self-regulated learning in higher education: structural model and indications drawn from expert interviews. *Studies in Higher Education* 40, 3 (2015), 454–470. https://doi.org/10.1080/03075079.2015.1004236

[12] Carol Dweck. 2012. *Mindset: Changing the way you think to fulfil your potential.* Hachette, London.

[13] Jacquelynne S. Eccles and Allan Wigfield. 2020. From expectancy-value theory to situated expectancy-value theory: A developmental, social cognitive, and sociocultural perspective on motivation. *Contemporary Educational Psychology* 61 (2020), 101859. https://doi.org/10.1016/j.cedpsych.2020.101859

[14] Ndudi O. Ezeamuzie and Jessica S. C. Leung. 2022. Computational Thinking Through an Empirical Lens: A Systematic Review of Literature. *Journal of Educational Computing Research* 60, 2 (2022), 481–511. https://doi.org/10.1177/07356331211033158 arXiv:https://doi.org/10.1177/07356331211033158

[15] Shuchi Grover and Roy Pea. 2013. Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher* 42, 1 (2013), 38–43. https://doi.org/10.3102/0013189X12463051 arXiv:https://doi.org/10.3102/0013189X12463051

[16] Ernesto Panadero. 2017. A Review of Self-regulated Learning: Six Models and Four Directions for Research. *Frontiers in psychology* 8 (2017), 422. https:

//doi.org/10.3389/fpsyg.2017.00422

[17] Erin E. Peters-Burton, Timothy J. Cleary, and Susan G. Forman. 2015. Professional development contexts that promote self-regulated learning and content learning in trainees. In *Self-regulated learning interventions with at-risk youth: Enhancing adaptability, performance, and well-being*. American Psychological Association, Washington, DC, US, 205–227. https://doi.org/10.1037/14641-010

[18] 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

[19] Seifodin Rajabi. 2012. Towards Self-Regulated Learning in School Curriculum. *Procedia - Social and Behavioral Sciences* 47 (2012), 344–350. https://doi.org/10.1016/j.sbspro.2012.06.661

[20] Peter Jacob Rich, Brian Jones, Olga Belikov, Emily Yoshikawa, and McKay Perkins. 2017. Computing and Engineering in Elementary School: The Effect of Year-long Training on Elementary Teacher Self-efficacy and Beliefs About Teaching Computing and Engineering. *International Journal of Computer Science Education in Schools* 1, 1 (2017), 1. https://doi.org/10.21585/ijcses.v1i1.6

[21] Katrin Saks and Äli Leijen. 2014. Distinguishing Self-directed and Self-regulated Learning and Measuring them in the E-learning Context. *Procedia - Social and Behavioral Sciences* 112 (2014), 190–198. https://doi.org/10.1016/j.sbspro.2014.01.1155

[22] Dale H. Schunk and Jeffrey A. Greene (Eds.). 2018. *Handbook of self-regulation of learning and performance* (2nd edition ed.). Routledge Taylor & Francis Group, New York and London.

[23] Yoonhee Shin and Donggil Song. 2022. The Effects of Self-Regulated Learning Support on Learners' Task Performance and Cognitive Load in Computer Programing. *Journal of Educational Computing Research* 60, 6 (2022), 1490–1513. https://doi.org/10.1177/07356331211052632 arXiv:https://doi.org/10.1177/07356331211052632

[24] Ummu Sholihah and Alfi Inayatul Firdaus. 2023. Student's Computational Thinking Ability in Solving Trigonometry Problems in the Review of Self-Regulated Learning. *Jurnal Penelitian Pendidikan IPA* 9, 2 (2023), 626–633. https://doi.org/10.29303/jppipa.v9i2.2821

[25] Valerie J. Shute, Chen Sun, and Jodi Asbell-Clarke. 2017. Demystifying computational thinking. *Educational Research Review* 22 (2017), 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

[26] 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

[27] L. Suchman. 1987. *Plans and Situated Actions: The Problem of Human Machine Communication*. Cambridge University Press, Cambridge.

[28] Ellen L. Usher and Dale H. Schunk. 2018. Social cognitive theoretical perspective of self-regulation. In *Handbook of self-regulation of learning and performance*, Dale H. Schunk and Jeffrey A. Greene (Eds.). Routledge Taylor & Francis Group, New York and London, 19–35.

[29] David Weintrop, Elham Beheshti, Michael Horn, Kai Orton, Kemi Jona, Laura Trouille, and Uri Wilensky. 2016. Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology* 25 (02 2016). https://doi.org/10.1007/s10956-015-9581-5

[30] Christiane Wesarg-Menzel, Rutmer Ebbes, Maud Hensums, Eline Wagemaker, Martina S. Zaharieva, Janneke P.C. Staaks, Alithe L. van den Akker, Ingmar Visser, Machteld Hoeve, Eddie Brummelman, Tycho J. Dekkers, Jaap A. Schuitema, Helle Larsen, Cristina Colonnesi, Brenda R.J. Jansen, Geertjan Overbeek, Hilde M. Huizenga, and Reinout W. Wiers. 2023. Development and socialization of self-regulation from infancy to adolescence: A meta-review differentiating between self-regulatory abilities, goals, and motivation. *Developmental Review* 69 (2023), 101090. https://doi.org/10.1016/j.dr.2023.101090

[31] Jeannette M. Wing. 2006. Computational Thinking. *Commun. ACM* 49, 3 (mar 2006), 33–35. https://doi.org/10.1145/1118178.1118215

[32] Philip H. Winne and Nancy E. Perry. 2000. Measuring Self-Regulated Learning. In *Handbook of Self-Regulation*. Elsevier, 531–566. https://doi.org/10.1016/B978-012109890-2/50045-7

[33] Christopher A. Wolters and Sungjun Won. 2018. Validity and the Use of Self-Report Questionnaires to Assess Self-Regulated Learning. In *Handbook of self-regulation of learning and performance*, Dale H. Schunk and Jeffrey A. Greene (Eds.). Routledge Taylor & Francis Group, New York and London, 307–322.

[34] Gülçin Zeybek. 2022. The Relationship between Pre-Service Teachers' Computational Thinking Skill Levels and Online Self-Regulated Learning Levels. *Critical Questions in Education* 13, 3 (2022), 264–288.

[35] Qian Zhou, Xiande Luo, Yilin Dong, Faiza Khalid, and Hongliang Ma. 2022. The Effectiveness of Self-Directed Learning in Developing Computational Thinking of Middle School Students. In *Proceedings of the 14th International Conference on Education Technology and Computers*. ACM, New York, NY, USA, 403–406. https://doi.org/10.1145/3572549.3572613