# How do Swedish technology teachers assess programming education in grade 4-6?

Eva-Lena Bjursten, Mälardalen University
Correspondence: eva-lena.bjursten@mdu.se
Lena Gumaelius, Royal Institute of Technology

## ABSTRACT

This study examines Swedish teachers' teaching and assessment practices in programming education for students in grades 4-6, with a focus on the technology subject. It investigates whether existing governing documents provide sufficient guidance for effective teaching and assessment in programming, particularly regarding Pedagogical Content Knowledge (PCK). The study addresses challenges faced by teachers, including limited training and a lack of instructional guidelines, stressing the importance of bridging this gap to support effective programming instruction and assessment. It explores assessment practices in programming within the technology subject, referring to previous studies that identify various approaches. The discussion includes product and process criteria for assessing programming tasks and emphasizes the need for clearer links between programming assessment and core technology content. The methodology involves semi-structured interviews with experienced teachers who taught programming prior to its inclusion in the curriculum. Analyzing the interview data helps examine alignment between teachers' assessment practices and governing documents. Results and discussion focus on one teacher, Camilla, with six years of programming teaching experience. It describes how Camilla facilitates curriculum goals and aligns assessments with grading criteria. The article also summarizes specific areas assessed in programming education and compares Camilla's criteria with essential content knowledge from previous studies. Based on the findings, the study concludes that while Camilla demonstrates comprehensive understanding of assessing programming knowledge, improvements are necessary in primary school programming education in Sweden. The existing governing documents inadequately support effective programming instruction, particularly in terms of content knowledge. It suggests identifying key characteristics of quality programming education at each stage of compulsory schooling and engaging in discussions to establish a strong educational foundation.

*Key Words: Computer programming, PCK, assessment, teacher education, professional development*

## 1. INTRODUCTION

In today's technology-driven world, programming education has become crucial for preparing students for the digital age. In Sweden, programming was introduced as new content in technology and mathematics subjects in compulsory school (Statens Skolverk, 2017). However, despite the curriculum update in 2022, the content and approach to teaching programming were not significantly altered (Statens Skolverk, 2022b). Nevertheless, many teachers face challenges in teaching and assessing this relatively new content, often due to limited training in the field (Vinnervik, 2020). Additionally, there is a lack of instructional guidelines available to assist teachers in approaching this relatively new content (Nordén, Heintz, Mannila, Parnes, & Regnell, 2017). Bridging this gap is essential to support teachers in effectively teaching and assessing programming. This article aims to explore the teaching and assessment practices of Swedish teachers in programming education, specifically focusing on the technology subject for students in grades 4-6, with a particular emphasis on Pedagogical Content Knowledge (PCK) in programming.

Assessment in programming within the technology subject has been previously studied, but different approaches have been taken. Björklund and Nordlöf (2021) identified two types of criteria used by teachers in assessing programming: product criteria and process criteria. Their findings suggest that process criteria, including investigative work, inventiveness, ability to use models, and capacity for self-assessment, could be useful in assessing students struggling with programming tasks. On the other hand, Mannila, Heintz, Kjällander and Åkerfeldt (2020) aimed to develop a comprehensive assessment framework for programming education across different grades. Their pilot study highlighted the effectiveness of combining multiple-choice and open-ended skills questions to assess students' abilities and identify misconceptions. However, there remains a need to establish clearer links between programming assessment and the core content of technology. Additionally, previous studies have indicated challenges in articulating the assessment of programming tasks, emphasizing the importance of assessing the entire process from idea to final product (Bjursten, Hartell, & Gumaelius, 2022).

Teaching and assessment are intricately linked, collectively shaping the learning experience. By examining how teachers assess their students' programming abilities, we can gain insights into the perceived importance and challenges associated with programming education. In this study, we delve into the assessment practices of an experienced teacher who taught programming even before its official inclusion in the curriculum. Through an interview with this teacher, we aim to investigate whether the existing governing documents provide sufficient guidance for teachers to effectively teach and assess their students in programming. The research question that forms the basis of this study is as follows:

Do the existing governing documents provide sufficient guidance for teachers to teach and assess their students in programming?

### *1.1. PCK in programming*

The teacher's ability to effectively teach programming is often referred to as Pedagogical Content Knowledge (PCK) in the literature (Shulman, 1987, 2013). While PCK is well-established in various school subjects (Doyle, Seery, Gumaelius, Canty, & Hartell, 2019), its clear definition in programming is still evolving (Hubbard, 2018). Saeli, Perrenet, M.G. Jochems and Zwaneveld (2010) conducted a study defining desirable PCK in programming for secondary school and higher education, which serves as a foundation for this current study focusing on programming education in grades 4-6 of compulsory schools. It is important to acknowledge that the concepts and approaches designed for higher education may not be directly applicable or suitable for younger students. By examining an experienced programming teacher's assessment practices and their alignment with the governing documents, as well as comparing them to the specific context of programming from Saeli's study, this investigation aims to provide valuable insights into the knowledge and skills necessary for effectively teaching programming concepts in this age group.

### *1.2. Programming in Steering Documents in Swedish Schools*

The inclusion of programming in the curriculum of Swedish schools is reflected in the steering documents that outline the objectives and guidelines for the technology subject. According to the syllabus, one of the core contents is the ability to program an object and control their own constructions or other objects through programming. This directive is stated under the section titled ‘Working methods for developing technical solutions’ (Statens Skolverk, 2022b, p. 259).

Furthermore, the syllabus is further supplemented by material on digitalization from the national education agency (Statens Skolverk, 2022a, pp. 9-10), which emphasizes that programming should be approached from a broader perspective beyond mere coding. This additional material highlights the need to consider programming as an integral part of digitalization, encompassing various aspects beyond the technical aspects of coding. However, despite the curriculum update in 2022, the grading criteria and goals were not significantly altered (Statens Skolverk, 2022b), resulting, again, in programming not being explicitly mentioned in the grading criteria and goals. As these steering documents play a crucial role in providing guidance to teaching, it is important to examine whether the existing documents offer sufficient support and clarity for teachers in assessing their students’ abilities programming in technology.

## 2. METHOD

The first author conducted semi-structured interviews with seven participants. All participants had previously participated in a study that included 14 experienced technology teachers in grades 4-6 taken place in 2018 and 2019 and were willing to participate again. The participants had been teaching programming before it became part of the curriculum, hence they are experienced teachers in programming in technology (Bjursten, Nilsson, & Gumaelius, 2022).

The aim of the interviews was both to follow up the data from a previous study and to focus even more on how assessment in programming is conducted.

The interviews were conducted in Zoom and lasted approximately one hour. The interviews were recorded on Zoom with the participants' consent and transcribed verbatim for analysis.

The present article provides an example of one of the seven teachers, Camilla, in which the teacher's assessment practices are described. Camilla has been teaching for 28 years and has been teaching programming in technology for the past six years. Her motivations stem from both the need to stay updated with the curriculum and her genuine passion for digital technology. This teacher was selected randomly from the material. The interview data was thematized according to how the teacher expressed assessment based on the goals and grading criteria outlined in the curriculum. The first and second authors then compared and discussed the findings to reach a consensus.

In the analyzing phase, the themes were compared with the study focuses on how the teacher's PCK aligns with the big ideas of programming as described in the literature Saeli et al. (2010).

## 3. RESULT AND DISCUSSION

This section comprises three subsections: 'Overview of the teaching and assessment practice', 'How the teacher aligns the assessment process to the grading criteria curriculum, and 'Summary of what is assessed in programming in the subject technology'. These subsections provide a comprehensive analysis of the teaching methods, their alignment with the assessment process, and a summary of the specific areas assessed in programming education.

### *3.1. Overview of the teaching and assessment practice*

In order to gain insight into Camilla's teaching practices and methods to promote the development of skills in her students, we first asked how she facilitates the acquisition of the three goals described in the technology syllabus in her teaching of programming.

When discussing the first goal, the capacity to carry out technological development and construction work (Statens Skolverk, 2022b), she mentions that they have built a LEGO robot, which involves both mechanical construction and programming. They need to follow instructions and troubleshoot if the robot doesn't function as intended.

For the second goal, knowledge of technical solutions and how their components interact to achieve purposefulness and functionality (Statens Skolverk, 2022b). In the context of programming, Camilla highlights the importance of synchronizing different parts of the program and understanding their functionality. She mentions examples like pedometers and timers, where it is essential to differentiate between them and ensure they work together without conflicts.

The third goal in the curriculum is to reflect on different choices of technical solutions and their consequences, as well as how technology changes over time (Statens Skolverk, 2022b). Here Camilla provides an example of working with smartwatches and comparing them to analog wristwatches, where students learn to program Micro:bit devices to perform multiple functions. They explore the advantages and disadvantages of different technical solutions and consider why it might be interesting to add additional features or functionality. Overall, Camilla emphasizes the importance of hands-on construction, programming, and critical thinking skills in relation to technology education.

### *3.2. How the teacher aligns the assessment process to the grading criteria curriculum*

The grading criteria of the Swedish curriculum for grade A at the end of Year 6 are in line with the goals which means that it is not necessary to separate when to assess the goal or the grading criteria. They include the ability to provide examples of technical solutions and describe their advantages, disadvantages, and how they have evolved over time. Also, the students are expected to investigate technical solutions and demonstrate an understanding of how different parts interact to achieve effectiveness and functionality. Additionally, they should be able to carry out simple technical development and construction work, formulating and selecting appropriate action options and documenting their solutions clearly (Statens Skolverk, 2022b).

During the interview, Camilla discussed her approach to supporting students in their learning process. She emphasized the importance of striking a balance between independent work and seeking guidance. Camilla clarified that the level of independence demonstrated by students influences their grades. The more independently students work and the fewer questions they need to ask, the higher their grade tends to be. However, she also made it clear that asking questions and seeking advice are important aspects of the learning process. Camilla stressed that it is through questioning and guidance that students can expand their understanding and knowledge.

Camilla further explained her assessment criteria for programming tasks using Micro:bit. She differentiated between different levels of proficiency based on the complexity of students' programs. The use of ready-made blocks in Micro:bit indicated a basic level of proficiency, while the inclusion of additional variables and functions elevated the work to higher levels. Camilla emphasized the importance of understanding the idea behind the program and encouraged students to experiment and revise their ideas, even if the program was not flawless.

She highlighted the significance of students formulating their own ideas and taking initiative in their work. Students who demonstrate a clear idea for a technical solution and can articulate it effectively are more likely to achieve higher grades. Camilla mentioned examples such as creating a clock or a step counter with specific functions, where students need to consider the different parts and actions required for the solution to work. The ability to demonstrate a thoughtful approach and effectively communicate their ideas in the program itself is considered when assessing students' work.

During the interview, Camilla shared her insights into the factors that contribute to higher grades in the context of technical development and construction work. She emphasized that while independent work and problem-solving skills are valued, asking for assistance and seeking guidance are not discouraged. Camilla recognized that learning occurs through questioning and encouraged students to ask for help when needed.

Camilla also mentioned that the length or complexity of a program does not necessarily determine the grade. Instead, she emphasized the importance of clarity, understanding, and the intention behind the program. As long as the core idea is functional and the purpose is evident, the program can meet the grading criteria.

### *3.3. Summary of what is assessed in programming in the subject technology*

Overall, the interview with Camilla highlighted her approach to meeting the goals and grading criteria of the curriculum. Her instructional strategies emphasized critical thinking, independent problem-solving, collaboration, and reflection. By engaging students in practical activities, encouraging experimentation, and providing guidance, when necessary, Camilla aimed to facilitate their development of technical knowledge, construction skills, and proficiency in programming.

In order to gain insight into how effective Camilla's assessment method is particularly in relation to essential content knowledge (CK), we have conducted a mapping exercise to determine how Camilla's assessment criteria align with the essential content knowledge identified by Saeli et al. (2010).

The resulting table illustrates that Camilla predominantly incorporates four of the eleven concepts and fundamental principles put forth by Saeli et al. (2010). Furthermore, while ‘Procedures’ are not explicitly mentioned, it is evident that she understands the concept based on her acknowledgement that a decomposed program is easier to debug, indicating familiarity with the concept without explicitly using its name. For example, the following interview extract serves as an illustration of Camilla's incorporation of some of these significant ideas:

> I usually say that the program should have three parts for a C, but there are no specific length requirements [length of the programming code], they [the functions/modules/parallel processes] just shouldn't clash with each other. And then, for the next level, there should be an additional variable. And at the A level, I usually require them to have a function of some kind... that calls another function. Alternatively, they can have multiple different variables. That can also compensate. So, I have found some sort of model for that, and I've managed to come up with student examples for each way so that I can show the students in advance what it should look like. I usually don't have a strict requirement for it [the program] to work perfectly, as long as one part works and I can understand the idea behind the others, it can be enough.

*Table 1.*
*Content Knowledge according to Saeli et al. (2010); Saeli, Perrenet, M.G. Jochems and Zwaneveld (2012)*

| Programming concepts | Camilla |
|---|---|
| Control structures: Loops, conditions and sequential | (X) |
| Functions, procedures, methods | X |
| Algorithms | |
| Variables (and Constants) | X |
| Parameters | |
| Data structure | |
| Decomposition | X |
| Reusability | |
| Arrays | |
| Logical thinking | |
| Formal language grammar and syntax | |

However, it is worth noting that certain foundational programming concepts are absent from Camilla's student assessments. Notably, the control structure loop is not explicitly addressed. This oversight potentially implies that students may lack crucial components essential for their future progression in programming. Without acquiring the necessary foundational knowledge, advancing to more complex stages of their programming education become significantly more challenging. An analogy can be drawn to the teaching of mathematics, wherein omitting one of the four fundamental counting methods hinders students when confronted with intricate problem-solving tasks. Similarly, the omission of a key concept like ‘loop’ at an early stage may render students ill-equipped to tackle and comprehend more intricate programming assignments. These observations find support in previous literature, further reinforcing this argument. However, since secondary and higher education teachers are included in Saeli et al. (2010), it is not clear which concepts of the above mentioned still apply to students in grades 4-6. Additionally, as an example, when utilizing a visual programming language with fixed syntax, it becomes necessary to explore alternative methods for evaluating 'Formal language grammar and syntax' since the syntax cannot be modified.

## 4. CONCLUSION

In conclusion, the teacher Camilla possesses a strong sense of confidence in her capacity as a programming instructor and demonstrates a comprehensive understanding of how to assess her students' programming knowledge. However, our findings highlight the need for improvements in programming education in primary schools in Sweden. The current descriptions and guidelines provided by governing documents are insufficient in supporting teachers to deliver effective programming instruction when it comes to content knowledge. To ensure a comprehensive and progressive approach to programming education, it is crucial to identify the key characteristics (e.g., ‘big ideas’) of good programming education in technology in each of the stages in compulsory schools. By engaging in discussions and exploring the relevant knowledge and skills required for programming at an early age, we can lay a strong foundation for students' educational advancement. This not only ensures compliance with legal requirements but also facilitates their overall learning and development.

Consequently, the next logical step could be to conduct a survey among experienced and accomplished teachers to gather their perspectives on the key elements (CK) that characterize good programming education in primary schools in technology. That would help identify the specific programming content experienced technology teachers use in primary school years 4-6 and thus contribute to the most effective in teaching programming to primary school students. By gaining insights from experienced teachers, we can enhance the quality of programming education in technology and provide valuable guidance for curriculum development and teacher training in this field.

## 5. REFERENCES

Bjursten, E.-L., Hartell, E., & Gumaelius, L. (2022, 7-10 Dec 2022.). *Assessment Practices in Computer Programming* 11th Biennial International Design and Technology Teacher’s Association

Research Conference (DATTArc). 7-10 Dec 2022., Southern Cross University, Gold Coast Campus, QLD. / [ed] Kurt Seemann and P John Williams. , 2022.

Bjursten, E.-L., Nilsson, T., & Gumaelius, L. (2022). Computer programming in primary schools: Swedish Technology Teachers' pedagogical strategies. *International Journal of Technology and Design Education*. https://doi.org/10.1007/s10798-022-09786-7

Björklund, L. E., & Nordlöf, C. (2021). Teacher's Assessment in Programming: Comparing Teachers' Individual Judgement Criteria in a Programming Course. *Techne serien-Forskning i slöjdpedagogik och slöjdvetenskap*, *28*(2), 188-195. https://journals.oslomet.no/index.php/techneA/article/view/4332

Doyle, A., Seery, N., Gumaelius, L., Canty, D., & Hartell, E. (2019). Reconceptualising PCK research in D&T education: Proposing a methodological framework to investigate enacted practice. *International Journal of Technology and Design Education*, *29*(3), 473-491.

Hubbard, A. (2018). Pedagogical content knowledge in computing education: A review of the research literature. *Computer Science Education*, *28*(2), 117-135.

Mannila, L., Heintz, F., Kjällander, S., & Åkerfeldt, A. (2020). Programming in primary education: Towards a research based assessment framework. Proceedings of the 15th Workshop on Primary and Secondary Computing Education, Virtual Event Germany.

Nordén, L.-Å., Heintz, F., Mannila, L., Parnes, P., & Regnell, B. (2017). Introducing Programming and Digital Competence in Swedish K–9 Education. In V. Dagienė & A. Hellas (Eds.), *10th International Conference on Informatics in Schools: Situation, Evolution, and Perspective, ISSEP 2017* (pp. 1-12). Springer Nature. https://doi.org/10.1007/978-3-319-71483-7

Saeli, M., Perrenet, J., M.G. Jochems, W., & Zwaneveld, B. (2010). *Portraying the pedagogical content knowledge of programming—The technical report*. http://teachingprogramming.esoe.nl/TechnicalReport/SPJZ/TechnicalReport.pdf.

Saeli, M., Perrenet, J., M.G. Jochems, W., & Zwaneveld, B. (2012). Programming: Teachers and Pedagogical Content Knowledge in the Netherlands. *Informatics in Education*, *11*(1), 81-114.

Shulman, L. S. (1987). Knowledge and Teaching: Foundations of the New Reform. *Harvard Educational Review*, *57*(1), 1-22.

Shulman, L. S. (2013). Those who Understand: Knowledge Growth in Teaching. *Journal of Education*, *193*(3), 1-11. https://doi.org/10.1177/002205741319300302

Statens Skolverk. (2017). *Läroplan för grundskolan, förskoleklassen och fritidshemmet 2011 Reviderad 2017 [Curriculum for the compulsory school, preschool class and school-age educare 2011 (revised 2017)]* https://www.skolverket.se/publikationsserier/styrdokument/2017/laroplan-for-grundskolan-forskoleklassen-och-fritidshemmet-2011-reviderad-2017?id=3813

Statens Skolverk. (2022a). *Få syn på digitaliseringen på grundskolenivå – Ett kommentarmaterial till läroplanerna förförskoleklass, fritidshem och grundskoleutbildning [Looking at digitisation at primary school level - A commentary on the curricula for pre-school, after-school and primary education]*.

Statens Skolverk. (2022b). *Läroplan för grundskolan, förskoleklassen och fritidshemmet 2022 [Curriculum for the compulsory school, preschool class and school-age educare 2022]*

Vinnervik, P. (2020). Implementing programming in school mathematics and technology: teachers' intrinsic and extrinsic challenges. *Int J Technol Des Educ* (32), 213–242. https://doi.org/10.1007/s10798-020-09602-0