

# Making programming part of teachers' everyday life – Programming affordances and constraints for K-12 mathematics and technology

Niklas Humble and Peter Mozelius  
*Department of Computer and System Science,  
Mid Sweden University – Campus Östersund, Östersund, Sweden*

## Abstract

**Purpose** – The conducted examination of programming affordances and constraints had the purpose of adding knowledge and value that facilitate the on-going national curricula revision; knowledge that also could be of general interest outside the Swedish K-12 context.

**Design/methodology/approach** – With a qualitative approach, the study was conducted as a document analysis where submitted lesson plans were the base for a directed content analysis.

**Findings** – This study presents findings on how the involvement of programming in mathematics and technology have potential to foster engagement and motivation among students. Findings also indicate that the implementation of programming can develop important general skills that go beyond the boundaries of mathematics and technology. Moreover, the identified constraints could be valuable to improve the on-going curriculum development for K-12 mathematics and technology.

**Research limitations/implications** – This qualitative study was conducted on a relatively small number of teachers where the majority has taken the courses on a voluntary basis. An important complement would be to conduct a larger quantitative study with data from a more general sample of K-12 teachers.

**Practical implications** – Results and discussions provide guidance for K-12 teachers and other stakeholders who want to introduce programming as a complementary tool in teaching and learning activities.

**Social implications** – The study has a contribution to the on-going implementation of the Swedish national curricula for K-12 mathematics and technology.

**Originality/value** – During the last years, many studies have been published on teacher training in programming, and how the training can be improved. This study goes beyond the actual teacher training and examine aspects teachers translate to theirs daily work after completing the training.

**Keywords** Programming, Lesson planning, Affordances, K-12 education, Mathematics, Technology

**Paper type** Research paper

## 1. Introduction

During recent years, programming has been integrated in kindergarten to grade 12 (K-12) education around the world (Tikva and Tambouris, 2021; Lindberg *et al.*, 2019; Szabo *et al.*, 2019). Reasons provided for integration are the development of important 21st century skills, and the need of science, technology, engineering and mathematics (STEM) professionals on future labour market (Tikva and Tambouris, 2021; Admiraal *et al.*, 2019). However, several studies report challenges for integration (Tikva and Tambouris, 2021). The Royal Society (2017) report that computing education is “patchy and fragile” in the United Kingdom (UK), and that better support should be provided for teachers. Previous research reports that a



potential issue for integrating programming is the choices that must be made between different programming tools, instruction methods and activities (Garneli *et al.*, 2015).

Programming was, decided by the Swedish government, to be integrated in K-12 curriculum in 2017 (Heintz *et al.*, 2017). The revised curriculum introduced programming and related concepts, such as digital competence, algorithms and controlling physical artefacts, as interdisciplinary concepts (Heintz *et al.*, 2017). However, programming was also specifically addressed for the subjects of mathematics and technology (Heintz *et al.*, 2017). Previous research on successful technology integration in education, has pointed out the importance of good examples for use of the technology (Viberg *et al.*, 2020; Lindberg *et al.*, 2017). Through this study, the authors intend to provide and discuss examples of how programming can be used for K-12 mathematics and technology and in which aspects programming can support teaching and learning.

The aim of the study was to examine programming affordances and constraints for K-12 mathematics and technology that can be identified in teachers' lesson plans on programming. The study was guided by the following research questions:

- RQ1. What programming affordances for K-12 mathematics and technology can be identified in the lesson plans?
- RQ2. What constraints for programming in K-12 mathematics and technology can be identified in the lesson plans?

## 2. Affordances

Affordances are what the environment, in a broad sense, provides when an animal utilises it (Gibson, 2015; Bower and Sturman, 2015). For example, an affordance of a mailbox is to mail letters (Bower and Sturman, 2015). The theory of affordances has its roots in ecological psychology and the works of Gibson (Bower and Sturman, 2015; Stendal *et al.*, 2016). According to Gibson (2015), affordances contain information from both the animal and the environment since they are based in the relationship between them. The theory of affordances has, since Gibson, been applied in many fields, such as Human-Computer Interaction and Information Systems (Stendal *et al.*, 2016).

Important for the wider application of the theory of affordances is the work by Norman (1999), who further developed the theory and introduced the concepts of *perceived affordances*, *constraints* and *conceptual models*. An important distinction that Norman introduces is that affordances are all possible actions that are provided by the environment, while perceived affordances only entail those that are perceived as possible (Norman, 2013, p. 19; Stendal *et al.*, 2016). This distinction is important in the design of artefacts since what can be controlled is the primary concern for the designer, that is, the affordances that are perceived as possible (Norman, 1999).

According to Norman (2013, p. 125), constraints are what limits the affordances that are perceived as possible. They can be considered the opposite of affordances (Tiberius, 2019) and are typically based in either physical, logical, cultural or semantic constraints, or a combination of these (Norman, 2013, pp. 125–130). For example, how many fingers that can hold a handle is constrained by the physical size of the handle (Koutamanis, 2006) and the only meaningful way to ride a motorcycle is facing forward protected by the windshield (Norman, 2013, p. 129). Constraints can be used as clues for actions by a designer (Norman, 2013, pp. 129–131). However, many of them are not fixed and change with time and context (Norman, 2013, pp. 129–131).

Although Norman (1999, 2013) has an emphasis on the designer perspective of affordances, there are examples in research on information systems that show different perspectives on how affordances can be used. In a literature review, Stendal *et al.* (2016) identified two different perspectives on the use of affordances in the field of information

systems. Affordances can be seen as embedded in the artefact and intended by the designer (Stendal *et al.*, 2016). This resembles Norman's (1999) description of perceived affordances as possible to control by a designer. Affordances can however also be seen as emerging, that is not pre-existing in the artefact, but experienced and perceived through actions and use (Stendal *et al.*, 2016). This would rather be considered a user perspective on affordances and can, for example, be viewed in Leonardi's (2013) study on shared affordances.

### 3. Related research

Programming in K-12 education has been actualised in recent years due to integration of programming in K-12 curriculum across the world (Tikva and Tambouris, 2021; Lindberg *et al.*, 2019; Szabo *et al.*, 2019). There are various ways of introducing programming in K-12 education one example is to introduce programming as its own subject while another example is to integrate programming in existing subjects (Nouri *et al.*, 2020). Sweden is one of the countries that have chosen the latter approach and as of 2017 programming has been integrated in K-12 mathematics and technology (Heintz *et al.*, 2017).

A central figure in programming for educational use is Seymour Papert, by some referred to as the *father of educational computing* (Stager, 2016). In the late 1960s, a programming language for children, Logo, was created by Papert and his colleagues, where programming is used to visualise and engage with mathematics in a playful way (Stager, 2016). Papert further coined the concept of computational thinking (CT), which was later developed and popularised by Jeannette Wing (Lodi and Martini, 2021). Several definitions and frameworks have been developed for CT, but a common notion is that CT is a way of thinking and solving problems with, or inspired by, computer science and programming (Lodi and Martini, 2021; Grover and Pea, 2018). In the Swedish K-12 context, programming is primarily understood as a process of problem-solving (Nouri *et al.*, 2020). However, programming is often also related to CT as an umbrella term to incorporate a broader perspective, including for example creativity, democratic dimensions and simulation (Nouri *et al.*, 2020).

In a literature review, Hickmott *et al.* (2018) report that many studies on CT and mathematics in K-12 context involve mathematics but mainly focus on programming skills, rather than mathematical concepts. A study by Bråting and Kilhamn (2022) conclude that the bridging of mathematics and programming is weak in elementary school textbooks for mathematics, and not enhancing the exploration of ideas and concepts in mathematics. Similar results can be found in a study by Elicer and Tamborg (2022) on didactical materials for connecting mathematics to technology comprehension. However, the study also shows examples of conceptual and operational integration (Elicer and Tamborg, 2022). Weintrop *et al.* (2016) suggest taxonomy of CT for mathematics and science, including four categories to address the challenge of what form CT should take in mathematics and science classroom: system thinking practices, computational problem-solving practices, modelling and simulation practices, and data practices.

In a study where programming and robotics were integrated in mathematics with 93 students in 6th grade, it is reported that students' understanding of mathematics and computational concepts improved significantly (Sáez-López *et al.*, 2019). Results from a study with two 7th grade classes showed increased performance in both mathematics and computer science when they were taught through coding, animation, dance, art and music with Scratch (Shamir *et al.*, 2019). The study further shows that students' interest in STEM and related careers increased (Shamir *et al.*, 2019). Similar results can be found in a mixed-methods study with 132 elementary students, where programming had a positive effect on students' perceptions of STEM, computer science and careers in the fields (Tran, 2018).

In a teaching model on programming abilities for K-12 (Sun *et al.*, 2022), programming is addressed as cultivating students' operational and cognitive abilities. The study by Sun *et al.* (2022) further report that block programming tools are commonly used, that problem-based

and project-based learning are common method for teaching programming, and that blended and cooperative and game-based learning have become teaching trends for programming. Nouri *et al.* (2020) points out, in their interview study with grade 1–9 teachers, that both general and CT specific skills are developed when students engage with programming. The general skills relate to 21st century skills and digital competence and include language skills, cognitive skills, creative problem-solving, collaborative skills and attitudes (Nouri *et al.*, 2020).

Ung *et al.* (2022) highlight that a challenge for introducing CT in schools could be a lack of understanding for CT among teachers. The study further explores how this challenge could be addressed with an E-learning system to train teachers' CT skills (Ung *et al.*, 2022). Kong *et al.* (2018) identify several potential challenges for introducing programming in K-12 education. The study indicate that students perceive programming as more meaningful and that it has greater impact and provide greater self-efficacy, both in creativity and programming, if the students already have interests for programming (Kong *et al.*, 2018). Further, the study indicates that boys perceive programming as more interesting than girls do, and that higher grade students have lower self-efficacy in programming and perceive it as less meaningful (Kong *et al.*, 2018).

Using programming in K-12 could also be challenging due to misconceptions induced by programming. Swidan *et al.* (2018) state in a study with 145 students (age 7–17) that the top three programming misconceptions among students relate to the sequences of statements, use of variables and user input. The study further suggests that misconceptions induced by programming environments, such as scratch, should be targeted and countered as early as possible with concept-rich and diverse teaching and learning material (Swidan *et al.*, 2018). Results presented by Cui and Ng (2021), show that primary school students find it challenging to combine mathematics with learning CT. The study further discusses that a possible explanation for these challenges could be the differences between CT and mathematical thinking (Cui and Ng, 2021).

## 4. Method

The qualitative approach to research tends to focus more on words than on numbers (Bryman, 2016, p. 375) and strive to produce knowledge, descriptions and understandings of people's interactions with their environments, other people, or objects (Polkinghorne, 2010). This study has been conducted with a qualitative approach to produce knowledge about teachers' understandings of programming affordances and constraints for K-12 mathematics and technology, and the meanings attached to these understandings. Data collection, data analysis, ethical considerations and reflections on trustworthiness are described in separate sub-headings below.

### 4.1 Data collection

Document analysis can be used for reviewing, evaluating and selecting data for analysis and categorisation (Bowen, 2009; Labuschagne, 2003). Documents used for this study were lesson plans created by participants at three instances of a teacher training course on programming for K-12 mathematics and technology. Lesson plans were created as the final assignment in the course with the instruction to create lesson plans on programming that the teachers found suited for their teaching and learning activities. The assignment further contained instructions for including motivations and reflections in the lesson plans, as well as to draw on what they have learned during the course. Since the targeted group for the teacher training course was teachers in grade 7–12 mathematics and technology, the lesson plans reflect that age group of students.

The rationale for having this assignment as the last assignment in the teacher training course on programming was that it should provide the participants the opportunity to translate what they have learned during the course into practical teaching and learning

material. The instructions for the assignment were relatively open to provide the foundation for participants to combine what they have learned in the course with their own teaching experience. To maximise the take-aways from the course for each teacher, the assignment was submitted to the course's online forum, giving all teacher participants access to all lesson plans created in the course instance. All lesson plans created in the course instances were included in this study, a total of 31 lesson plans. 9 of the lesson plans were created in the autumn semester of 2020, 9 in the spring semester of 2021 and 13 in the autumn semester of 2021. The gender distribution of the teachers, that created the lesson plans were 13 females and 18 males.

#### 4.2 Data analysis

Directed content analysis with deductive-inductive approach was used to analyse the collected material to expand knowledge on teachers' use of programming for K-12 mathematics and technology (Hsieh and Shannon, 2005; Srnka and Koeszegi, 2007; Drisko and Maschi, 2016). The analysis started with the use of predetermined categories, based in the theory of affordances, in line with the idea of deductive coding (Graneheim *et al.*, 2017; Drisko and Maschi, 2016). However, the analysis also has elements of inductive coding, for example, in the clustering of super-categories (Graneheim *et al.*, 2017; Drisko and Maschi, 2016; Juhola *et al.*, 2007; Srnka and Koeszegi, 2007; Polley *et al.*, 2007). To strengthen the trustworthiness of the study, a detailed description of the coding approach is provided below, inspired by Assarroudi *et al.* (2018) steps for directed content analysis.

In the first step, a categorisation matrix was created in a spreadsheet document, based on the theory of Affordances, with two categories: *affordances* and *constraints*. In the second step, definitions for the categories were derived from theory and related research, which can be found in section 2 *Affordances*. In the third step, coding rules for the categories were decided to be codes addressing programming affordances or constraints for K-12 mathematics and technology. The categorisation matrix was then tested in the fourth step with a small sample of the collected lesson plans. The preliminary results, with the sample lesson plans, were discussed among the authors, and the categorisation matrix was deemed appropriate for further analysis. In the fifth step, anchor examples for the categories were specified through adding related research to the theory of Affordances, providing a richer definition of *affordances* and *constraints* (Assarroudi *et al.*, 2018; Mayring, 2000).

In the sixth step, the main data analysis was performed. Each lesson plan was thoroughly read, and extracts related to programming affordances or constraints for K-12 mathematics and technology were copied into either the category of affordances or constraints in the spreadsheet document. In the seventh step, inductive abstraction was used to move from preliminary codes to sub-categories. The codes were clustered in sub-categories in the spreadsheet document based on their similarities, meanings and differences. This step created a total of 24 sub-categories divided between the two categories *affordances* and *constraints*. To create more specified categories, the seventh step was repeated but this time with the sub-categories instead of the codes. Sub-categories were clustered into new categories (super-categories) based on their similarities, meanings and differences. This created five super-categories, replacing the previous categories of affordances and constraints, containing the same 24 sub-categories (Table 1). In the final eighth step, extracts from the collected lesson plans were chosen to be used in the presentation of the results (5 *Results and analysis*) and to show the link between collected lesson plans, codes and categories (Assarroudi *et al.*, 2018).

#### 4.3 Trustworthiness

To ensure quality in the content analysis, the following criteria for trustworthiness have been applied: credibility, dependability, confirmability, authenticity and transferability

Super-categories	Sub-categories
Programming to support subject content	Adaptation Repetition Re-use Visualisation Variety Creativity Power Generalisation Feedback
Programming to foster engagement and motivation	Mixing programming tools Fun activities Problem-based learning Curiosity
Programming to develop general skills	Error handling Deconstruction Independence Producer Holistic views
Organisation related constraints for programming	Computers Lack of time
Student related constraints for programming	Prior knowledge Resistance Focus Patience

**Table 1.**  
Summary of super-categories and sub-categories from the directed content analysis

(Kyngäs *et al.*, 2020; Elo *et al.*, 2014). The study strives for trustworthiness regarding credibility and dependability by clearly describing the process of data collection and data analysis (Kyngäs *et al.*, 2020; Graneheim and Lundman, 2004). To strengthen the study's trustworthiness regarding confirmability and authenticity, quotes from the collected material are presented in the analysis to show connection between data and interpretation of data (Kyngäs *et al.*, 2020). Transferability is to a large extent about the readers' ability to determine the application of results to relevant fields and contexts (Kyngäs *et al.*, 2020; Elo *et al.*, 2014). The process described in previous criteria for trustworthiness, together with discussion of results, are steps taken to ensure transferability. Authors are course facilitators in the course instances where data have been collected and are experienced in research and teaching of the studied topic.

#### 4.4 Ethical considerations

An important principle in qualitative research is that of informed consent (Shaw, 2003). Participants were informed about intent to use course material for research and that consent could be withdrawn at any point without repercussions. Information have been provided verbally in teaching and learning sessions and written in online course forums. To ensure confidentiality and privacy, another important principle in qualitative research (Pietilä *et al.*, 2020; Shaw, 2003), no names or personal details about participants are shared in the research. Quotes from the lesson plans have been translated to English and to some extent rewritten, without changing the underlying meaning, to protect participants confidentiality and privacy, and for readability. Since authors were course facilitators, the principle of practitioner research (Shaw, 2003) is relevant. To ensure participants give consent freely, data have been collected from concluded courses where participants had a chance to retrieve final course grades.

## 5. Results and analysis

In this section, results and analysis from the study is presented in sub-headings. In the first sub-heading, identified affordances related to using programming for supporting teaching and learning of subject content in mathematics and technology are presented. In the second sub-heading, identified affordances related to using programming for fostering student engagement and motivation are presented. In the third sub-heading, identified affordances related to using programming for supporting development of general skills are presented. In the fourth sub-heading, identified constraints (organisation related and student related) for using programming in mathematics and technology are presented. Quotes and examples from the lesson plans presented in the sub-headings have been translated from Swedish to English and, to some extent, rewritten for readability. The underlying meaning of the quotes and examples has not been changed.

### 5.1 Programming for subject support

In the collected lesson plans, teachers use programming to support mathematics and technology through adaptation, repetition and re-use of subject content. Subject contents are adapted to different levels of difficulty with programming instructions to suite different student groups. For example, more advanced functionality of programs is included or excluded depending on prior knowledge of students. Lesson plans reflect that, teachers use programming to repeat subject content for better learning and for specific repetition in subject content. For example, first solving a problem manually and then with programming, and by repeating mathematical calculations with loops instead of manual repetition. Programming material in the lesson plans are further used and re-used by teachers in different subjects, but also as a foundation for interdisciplinary collaborations, for example between mathematics and technology.

Teachers use programming in the lesson plans to support mathematics and technology through visualisation, variety and creativity of subject content. Teachers use visual programming environments, such as scratch, in the lesson plans to support subject content, but also to lower the threshold of programming activities. Teachers use Python programming libraries, such as Turtle, in the lesson plans to visualise subject content, for example mathematics and geometry (Example 1). Lesson plans reflect that, teachers use programming for variety in their teaching and learning activities. With programming, students get the opportunity to work with similar content in a new context and to use their knowledge in a different way. Lesson plans further reflect that, teachers use programming to support creativity in subject content. One teacher expresses in a lesson plan: "I hope that this assignment also will give students the opportunity to be creative and find their own solutions".

#### *Lesson 9 – PythonTurtle*

- (1) *Draw with Python! If you want to draw with Python you have to create your program with PythonTurtle. You can choose to use that at repl.it.*
- (2) *Choose PythonTurtle for your new program and give the program a name. It will give you access to three windows: one for your code, one for the output and one for the turtle-drawing.*
- (3) *When you start your program, write: "import turtle as t" for the program to know you intend to draw something. You do not have to write "turtle" every time you call on turtle, it will be enough with "t" [ . . ]*
- (4) *Write a program for drawing a circle with PythonTurtle. You can decide the radius.*

*Example 1.* Lesson plan with Turtle library in Python to draw a circle.

Teachers support subject content of mathematics and technology in the lesson plans with the power, generalisation and feedback that programming can provide. Lesson plans indicate that, teachers use programming to support subject content that can be difficult to learn and teach without computer power. For example, problem-solving, statistics, prime numbers and number sequences are addressed with programming in lesson plans. Teachers also use programming and generalised programming solutions in the lesson plans to build on students' previous knowledge and learnings. Lesson plans reflect that, teachers use programming to generalise students' knowledge about problem-solving to other domains, by re-using and expanding algorithms that students develop to work with other subject content. Programming is also involved by teachers in the lesson plans to provide instant feedback on students' work, as a complement to teacher support. As expressed by one teacher in a lesson plan: "because flawed code instantly is showed by the program not working, the students get instant feedback on their work without asking for support. Therefore, I believe that programming can be a good opportunity to exercise students' autonomy".

### 5.2 Programming for motivation

Collected lesson plans reflect that, teachers use programming to foster student engagement and motivation for mathematics and technology through mixing different programming tools and enjoyable activities with programming. In the lesson plans, teachers use multiple programming tools, and it is expressed that mixing programming tools is perceived to support students' understanding and engagement: "I believe that the mixture of block programming and textual programming provides a greater understanding, and therefore also a greater engagement among the students". Teacher reflections in the lesson plans indicate that students find programming to be a fun activity, and that programming has the potential of boosting engagement and motivation for mathematics and technology. In relation to one lesson plan (Example 2), a teacher expressed the following: "The assignment with Micro:bit in technology has engaged my students in a good way and many thought that it was a fun and interesting assignment". It is further expressed in the lesson plans that, teachers find programming to be fun and that it can boost their own motivation for teaching: "I burn with eager to test this lesson plan [ . . . ] I believe that it can be really fun".

*What should you do?*

- (1) *You are going to program a micro:bit through the website below and test that your code works on a real micro:bit.*
- (2) *You are going to program some of the examples that you can find on [microbit.org](http://microbit.org)*
- (3) *The assignments are divided into different level of difficulty.*

*Assignment:*

- (1) *Go to the website [microbit.org](http://microbit.org), there you find information about micro:bit, but there is also a tab where you can write code. You can program with blocks, but you can also choose text. The text is written in the programming language Python.*

*Example 2.* Lesson plan with Micro:bit.

Lesson plans reflect that, teachers use programming to foster student engagement and motivation for mathematics and technology through problem-based learning. In the lesson plans, teachers use programming to incorporate problem-based learning in their teaching and learning activities. Reflections in the lesson plans show that teachers perceive programming with problem-based learning to be an effective approach for making subject content real

life-oriented for students: “I believe that problem-based learning in many ways is an effective approach, because it is real life-based, and therefore it quickly makes the benefits visible for the students”. Lesson plans further reflect that, teachers use programming to stimulate students’ curiosity in learning and develop free thinking. In one lesson plan, a teacher express: “I want my students to think about what in their environment are programmed. With the help of images, I hope they will start to reflect freely and that this will awaken some curiosity among them”.

### 5.3 Programming for general skills

Collected lesson plans reflect that, teachers use programming to support students’ development of general skills, beyond mathematics and technology, through error handling and deconstruction. In the lesson plans, teachers reflect on and use programming for supporting students to reflect on their work, try their way forward, detect errors and correcting them. A teacher expresses in one lesson plan: “I want students to use pseudo code, detect errors in the code, improve their programs, understand programs, and how to code programs”. Lesson plans indicates that teachers use programming as a tool for supporting students to deconstruct problems and assignments (Example 3). It is further reflected in the lesson plans that, teachers consider it important that, students should learn to structure and break down problems in smaller parts, to be able to methodically solve them.

#### *Lesson 2 – programming part 1.*

*My thought is that we start with a program where students can calculate the circumference of a triangle, based on the input of base and height.*

*Example of code:*

```
#Calculate circumference of a triangle
```

```
base = input("Base of the triangle: ")
```

```
b = int(base)
```

```
height = input("Hight of the triangle: ")
```

```
h = int(hight)
```

```
y = 2 * b + 2 * h
```

```
print("The circumference of your triangle is: ", y)
```

*Example 3.* Lesson plan where the calculation of a triangle’s circumference is deconstructed with Python code.

Lesson plans reflect that, teachers use programming to support students’ development of general skills, beyond mathematics and technology, through being independent and applying a producer perspective. Lesson plans reflect that, teachers consider being independent to be an important trait for students and that this can be developed with, for example, error handling. In one lesson plan, a teacher express: “Also, I hope that I teach students to work independently. That they can try, discover errors, try again and thereby grow. I want this work to train students to dare to try themselves and by their own discover error that they can correct”. Related to this, it is expressed that programming can support students to create their own solutions and technology, and not be dependent on what is provided for them. In one lesson plan, a teacher expresses that programming can make students experience the “satisfaction of being a producer and not only a consumer”.

General skills beyond mathematics and technology are also supported and developed by teachers in the lesson plans through holistic views with programming. That is, teachers use programming to connect subject content or the subject to something bigger for the students. For example, programming and functions are used to connect smaller parts of mathematics to something bigger and more whole at the end. In one lesson plan, a teacher expresses: “to put mathematical calculations in functions [...] in a bigger program, developing more and more knowledge and at the end, everything is put together”. Programming is further used by teachers in the lesson plans to connect subject content of technology to the outside world, by showing what and how programming is used in society. As one teacher explains in a lesson plan: “I want to start with what programming is, how programmed technology works in society and why it is important to developed knowledge about and understand programming”.

#### *5.4 Constraints for programming*

Besides affordances, collected lesson plans reflect teachers’ constraints in using programming for mathematics and technology. These constraints have been clustered in *organisation related constraints* and *student related constraints*. Organisation related constraints in the lesson plans reflect teachers’ concern for programming related to school computers and lack of time. In one lesson plan, a teacher expresses: “Unfortunately, we had to end the programming much earlier than I had hoped because of software problems that Python caused for our computers. Not the program [Python] in itself, but the installation service that the municipality uses did not play along with the program [Python] and rendered our computers unusable”. Lesson plans also reflect teachers’ perceived lack of time as a constraint for programming in mathematics and technology. However, teacher reflections in the lesson plans suggest that this could mainly be a constraint for mathematics, and that the time issue for programming is not as critical in technology.

Student related constraints in the lesson plans reflect teachers’ concerns for programming in relation students’ prior knowledge, resistance, focus and patience. The lesson plans reflect that, teachers perceive a wide range of prior knowledge in programming among students to be a potential constraint for programming in mathematics and technology. One teacher expresses the following in relation to the lesson plan (Example 4): “A potential problem with this lesson plan is that a reasonable experienced programmer or a good problem-solver can solve this assignment in a couple of minutes. This can create problems for the group and the learning. It demands that there are additional activities for these students to do”. A potential constraint reflected in the lesson plans, related to students’ prior knowledge, is that students could be resilient towards programming activities. In one lesson plan, a teacher suggests that a potential solution could be “[...] short and easy programming assignments that are recurring and keeps focus on the mathematics”. Lesson plans further reflect that, teachers perceive students’ lack in focus and patience to be potential constraints for programming. However, lesson plans also reflect that, teachers perceive programming to have the potential of supporting students’ development of these traits.

#### *Assignment 2 – Recursive number sequence*

*This assignment is appropriate for students that are familiar with the for-loop and should be conducted in small groups with tutoring from the teacher [...]. The tricky part of this assignment is that students should use the previous number in the sequence for calculating the next. Further, students should use the print()-function in the for-loop to examine how the number sequence change.*

A recursive number sequence is a number sequence that uses previous numbers to calculate the next. Sometimes a number sequence can be expressed both explicitly [...] and recursively. For example, the number sequence  $a_n = 2a_{n-1}$  can also be expressed  $a_1 = 1, a_2 = 2a_{n-1}$

- (1) Create a program that calculates the 10th element in the number sequence  $a_1 = 1, a_n = 2a_{n-1} + 1$ .
- (2) Express the recursive number sequence with an explicit formula.

*Example 4.* Lesson plan on recursive number sequences with Python

## 6. Discussion and conclusion

As pointed out in previous research, the existence of good examples is important when integrating new technology in education (Viberg *et al.*, 2020; Lindberg *et al.*, 2017). With Swedish and international integration of programming in K-12 education (Tikva and Tambouris, 2021; Lindberg *et al.*, 2019; Szabo *et al.*, 2019; Heintz *et al.*, 2017), this study has sought to provide examples of how programming can be used for K-12 mathematics and technology, and how teachers reflect on this. Several affordances and constraints have been identified in the analysis of collected lesson plans from K-12 teachers that have participated in a teacher training course on programming. Identified affordances relate to mainly three aspects of teaching and learning: *programming to support subject content*, *programming to foster engagement and motivation*, and *programming to develop general skills*. Identified constraints concerns mainly two aspects of teaching and learning: *organisation related constraints for programming* and *student related constraints for programming*.

Contrary to many other studies on programming and mathematics in K-12, which mainly focus on programming skills (Hickmott *et al.*, 2018), this study has focused on how subjects of mathematics and technology can be supported by programming activities. The study provides examples of how K-12 teachers implement programming in their subjects through lesson plans. Most of the identified affordances (Table 1) relate to how programming can support the subject content of mathematics and technology. Previous research has shown the positive effects programming can have on students' performance and interests in STEM (Shamir *et al.*, 2019; Tran, 2018), this study show that programming is perceived by the teachers to engage and motivate both students and teachers in mathematics and technology.

Like previous research (Sun *et al.*, 2022; Nouri *et al.*, 2020), this study has identified general skills that go beyond the subjects of mathematics and technology. These stress, for example, that students should be able to identify and correct errors, deconstruct problems, work independent, produce technology (and not only consume it) and view learning and school subjects holistically. However, as in previous research, this study has identified challenges or constraints for integrating programming in K-12 mathematics and technology. Constraints identified in this study, concerning students' prior knowledge, resistance, focus and patience; can be related to studies by Swidan *et al.* (2018) and Cui and Ng (2021), where it is shown that challenges and misconceptions can arise when programming is combined with another subject. This should be especially true with students that already find mathematics and technology challenging, adding programming might be overwhelming.

What has not been addressed in this study but deserves more attention is the challenge pointed out by Kong *et al.* (2018). They highlight that, students who are boys, younger and already have an interest in programming, perceive programming as more meaningful and show more self-efficacy (Kong *et al.*, 2018). Since programming and STEM have been identified as important skills for future labour market (Tikva and Tambouris, 2021; Admiraal *et al.*, 2019), it should also be important that these skills are accessible for all students to

develop. These are challenges teachers face and should also be addressed and supported in future research. This could for example be done through similar E-learning systems suggested by Ung *et al.* (2022) for supporting and training teachers' understanding of CT, to better introduce it in schools.

This study suggests that teachers that have participated in a teacher training course on programming perceive several programming affordances and constraints for K-12 mathematics and technology. Further, the study concludes that programming has the potential of supporting several aspects of teaching and learning in these subjects. Programming can support subject content, foster students' engagement and motivation and support the development of important skills that go beyond the boundaries of the subjects. However, for this to be possible and the integration of programming to be successful, the constraints teachers highlight needs to be addressed. Further, more research needs to be conducted that provide good examples of how programming can be used and implemented in teachers' everyday life. Several findings in this study can be related to previous research. However, with the use of the theory of affordances, this study has identified several hands-on and practical examples of what programming can provide for K-12 subjects' mathematics and technology, and what challenges teachers perceive for this. These findings can have real practical implications for teachers and other stakeholders working with programming or planning to integrate programming in K-12 mathematics and technology, since the findings provide good examples from teachers for teachers.

## 7. Limitations and future research

This study was conducted on lesson plans from three instances of a teacher training course at one university. Similar courses are given at various universities, and a larger nation-wide quantitative survey would be a valuable complement to the results presented in this study. Furthermore, the vast majority of the K-12 teachers in this study have taken these courses on a voluntary basis. The conclusions on the study would probably have been different with data gather from teachers that do not want to take these courses. As mentioned earlier, previous research has identified that younger male students who already have an interest in programming perceive programming as more meaningful than other students. The next planned study is to investigate which factors that can appeal to younger female students in learning activities for programming. This planned study will be carried out during 2022 as a part of the Erasmus + project *Gaming4Coding*.

## References

- Admiraal, W., Post, L., Guo, P., Saab, N., Makinen, S., Rainio, O., Vuori, J., Bourgeois, J., Kortuem, G. and Danford, G. (2019), "Students as future workers: cross-border multidisciplinary learning labs in higher education", *International Journal of Technology in Education and Science*, Vol. 3 No. 2, pp. 85-94, available at: <https://www.learntechlib.org/p/207262/> (accessed 15 July 2022).
- Assarroudi, A., Heshmati Nabavi, F., Armat, M.R., Ebadi, A. and Vaismoradi, M. (2018), "Directed qualitative content analysis: the description and elaboration of its underpinning methods and data analysis process", *Journal of Research in Nursing*, Vol. 23 No. 1, pp. 42-55, doi: [10.1177/1744987117741667](https://doi.org/10.1177/1744987117741667).
- Bowen, G.A. (2009), "Document analysis as a qualitative research method", *Qualitative Research Journal*, Vol. 9 No. 2, pp. 27-40, doi: [10.3316/QRJ0902027](https://doi.org/10.3316/QRJ0902027).
- Bower, M. and Sturman, D. (2015), "What are the educational affordances of wearable technologies?", *Computers and Education*, Vol. 88, pp. 343-353, doi: [10.1016/j.compedu.2015.07.013](https://doi.org/10.1016/j.compedu.2015.07.013).
- Bråting, K. and Kilhamn, C. (2022), "The integration of programming in Swedish school mathematics: investigating elementary mathematics textbooks", *Scandinavian Journal of Educational Research*, Vol. 66 No. 4, pp. 594-609, doi: [10.1080/00313831.2021.1897879](https://doi.org/10.1080/00313831.2021.1897879).

- Bryman, A. (2016), *Social Research Methods*, Oxford University Press, Oxford.
- Cui, Z. and Ng, O.L. (2021), "The interplay between mathematical and computational thinking in primary school students' mathematical problem-solving within a programming environment", *Journal of Educational Computing Research*, Vol. 59 No. 5, pp. 988-1012, doi: [10.1177/0735633120979930](https://doi.org/10.1177/0735633120979930).
- Drisko, J.W. and Maschi, T. (2016), *Content Analysis*, Oxford University Press, New York, NY.
- Elicer, R. and Tamborg, A.L. (2022), "Nature of the relations between programming and computational thinking and mathematics in Danish teaching resources", *Proceedings of the 15th International Conference on Technology in Mathematics Teaching (ICTMT 15)*, Danish School of Education, Aarhus University, p. 45.
- Elo, S., Kääriäinen, M., Kanste, O., Pölkki, T., Utriainen, K. and Kyngäs, H. (2014), "Qualitative content analysis: a focus on trustworthiness", *SAGE Open*, Vol. 4 No. 1, pp. 1-10, doi: [10.1177/2158244014522633](https://doi.org/10.1177/2158244014522633).
- Garneli, V., Giannakos, M.N. and Choriantopoulos, K. (2015), "Computing education in K-12 schools: a review of the literature", *2015 IEEE Global Engineering Education Conference (EDUCON)*, IEEE, pp. 543-551, doi: [10.1109/EDUCON.2015.7096023](https://doi.org/10.1109/EDUCON.2015.7096023).
- Gibson, J.J. (2015), *The Ecological Approach to Visual Perception*, Psychology Press, New York, NY.
- Graneheim, U.H. and Lundman, B. (2004), "Qualitative content analysis in nursing research: concepts, procedures and measures to achieve trustworthiness", *Nurse Education Today*, Vol. 24 No. 2, pp. 105-112, doi: [10.1016/j.nedt.2003.10.001](https://doi.org/10.1016/j.nedt.2003.10.001).
- Graneheim, U.H., Lindgren, B.M. and Lundman, B. (2017), "Methodological challenges in qualitative content analysis: a discussion paper", *Nurse Education Today*, Vol. 56, pp. 29-34, doi: [10.1016/j.nedt.2017.06.002](https://doi.org/10.1016/j.nedt.2017.06.002).
- Grover, S. and Pea, R. (2018), "Computational thinking: a competency whose time has come", in Sentence, S., Barendsen, E. and Schulte, C. (Eds), *Computer Science Education: Perspectives on Teaching and Learning in School*, Bloomsbury Academic, pp. 19-38.
- Heintz, F., Mannila, L., Nordén, L.Å., Parnes, P. and Regnell, B. (2017), "Introducing programming and digital competence in Swedish K-9 education", *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, Springer, Cham, pp. 117-128, doi: [10.1007/978-3-319-71483-7\\_10](https://doi.org/10.1007/978-3-319-71483-7_10).
- Hickmott, D., Prieto-Rodriguez, E. and Holmes, K. (2018), "A scoping review of studies on computational thinking in K-12 mathematics classrooms", *Digital Experiences in Mathematics Education*, Vol. 4 No. 1, pp. 48-69, doi: [10.1007/s40751-017-0038-8](https://doi.org/10.1007/s40751-017-0038-8).
- Hsieh, H.F. and Shannon, S.E. (2005), "Three approaches to qualitative content analysis", *Qualitative Health Research*, Vol. 15 No. 9, pp. 1277-1288, doi: [10.1177/1049732305276687](https://doi.org/10.1177/1049732305276687).
- Juhola, H., Kukkurainen, M.L. and Suominen, T. (2007), "Rheumatology nurses' work-related empowerment", *Musculoskeletal Care*, Vol. 5 No. 2, pp. 91-97, doi: [10.1002/msc.102](https://doi.org/10.1002/msc.102).
- Kong, S.C., Chiu, M.M. and Lai, M. (2018), "A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education", *Computers and Education*, Vol. 127, pp. 178-189, doi: [10.1016/j.compedu.2018.08.026](https://doi.org/10.1016/j.compedu.2018.08.026).
- Koutamanis, A. (2006), "Buildings and affordances", in Gero, J.S. (Ed.), *Design Computing and Cognition '06*, Springer, Dordrecht, pp. 345-364, doi: [10.1007/978-1-4020-5131-9\\_18](https://doi.org/10.1007/978-1-4020-5131-9_18).
- Kyngäs, H., Kääriäinen, M. and Elo, S. (2020), "The trustworthiness of content analysis", in Kyngäs, H., Mikkonen, K. and Kääriäinen, M. (Eds), *The Application of Content Analysis in Nursing Science Research*, Springer, pp. 41-48.
- Labuschagne, A. (2003), "Qualitative research: airy fairy or fundamental", *The Qualitative Report*, Vol. 8 No. 1, pp. 100-103, doi: [10.46743/2160-3715/2003.1901](https://doi.org/10.46743/2160-3715/2003.1901).
- Leonardi, P.M. (2013), "When does technology use enable network change in organizations? A comparative study of feature use and shared affordances", *MIS Quarterly*, Vol. 37 No. 3, pp. 749-775, available at: <https://www.jstor.org/stable/43825998>

- Lindberg, O.J., Olofsson, A.D. and Fransson, G. (2017), "Same but different? An examination of Swedish upper secondary school teachers' and students' views and use of ICT in education", *The International Journal of Information and Learning Technology*, Vol. 34 No. 2, pp. 122-132, doi: [10.1108/IJILT-09-2016-0043](https://doi.org/10.1108/IJILT-09-2016-0043).
- Lindberg, R.S., Laine, T.H. and Haaranen, L. (2019), "Gamifying programming education in K-12: a review of programming curricula in seven countries and programming games", *British Journal of Educational Technology*, Vol. 50 No. 4, pp. 1979-1995, doi: [10.1111/bjet.12685](https://doi.org/10.1111/bjet.12685).
- Lodi, M. and Martini, S. (2021), "Computational thinking, between Papert and wing", *Science and Education*, Vol. 30 No. 4, pp. 883-908, doi: [10.1007/s11191-021-00202-5](https://doi.org/10.1007/s11191-021-00202-5).
- Mayring, P. (2000), "Qualitative content analysis", *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research*, Vol. 1 No. 2, doi: [10.17169/fqs-1.2.1089](https://doi.org/10.17169/fqs-1.2.1089).
- Norman, D.A. (1999), "Affordance, conventions, and design", *Interactions*, Vol. 6 No. 3, pp. 38-43, doi: [10.1145/301153.301168](https://doi.org/10.1145/301153.301168).
- Norman, D. (2013), *The Design of Everyday Things: Revised and Expanded edition*, The MIT Press, Cambridge, MA.
- Nouri, J., Zhang, L., Mannila, L. and Norén, E. (2020), "Development of computational thinking, digital competence and 21st century skills when learning programming in K-9", *Education Inquiry*, Vol. 11 No. 1, pp. 1-17, doi: [10.1080/20004508.2019.1627844](https://doi.org/10.1080/20004508.2019.1627844).
- Pietilä, A.M., Nurmi, S.M., Halkoaho, A. and Kyngäs, H. (2020), "Qualitative research: ethical considerations", in Kyngäs, H., Mikkonen, K. and Kääräinen, M. (Eds), *The Application of Content Analysis in Nursing Science Research*, Springer, Cham, pp. 49-69, doi: [10.1007/978-3-030-30199-6\\_6](https://doi.org/10.1007/978-3-030-30199-6_6).
- Polkinghorne, D.E. (2010), "Qualitative research", in Thomas, J.C. and Hersen, M. (Eds), *Handbook of Clinical Psychology Competencies*, Springer, pp. 425-456, doi: [10.1007/978-0-387-09757-2\\_16](https://doi.org/10.1007/978-0-387-09757-2_16).
- Polley, M.J., Seers, H.E., Cooke, H.J., Hoffman, C. and Paterson, C. (2007), "How to summarise and report written qualitative data from patients: a method for use in cancer support care", *Supportive Care in Cancer*, Vol. 15 No. 8, pp. 963-971, doi: [10.1007/s00520-007-0283-2](https://doi.org/10.1007/s00520-007-0283-2).
- (The) Royal Society (2017), "After the reboot: computing education in UK schools", available at: <https://royalsociety.org/~media/events/2018/11/computing-education-1-year-on/after-the-reboot-report.pdf> (accessed 2 March 2022).
- Sáez-López, J.M., Sevillano-García, M.L. and Vazquez-Cano, E. (2019), "The effect of programming on primary school students' mathematical and scientific understanding: educational use of mBot", *Educational Technology Research and Development*, Vol. 67 No. 6, pp. 1405-1425, doi: [10.1007/s11423-019-09648-5](https://doi.org/10.1007/s11423-019-09648-5).
- Shamir, M., Kocherovsky, M. and Chung, C. (2019), "A paradigm for teaching math and computer science concepts in k-12 learning environment by integrating coding, animation, dance, music and art", *2019 IEEE Integrated STEM Education Conference (ISEC)*, IEEE, pp. 62-68, doi: [10.1109/ISECon.2019.8882015](https://doi.org/10.1109/ISECon.2019.8882015).
- Shaw, I.F. (2003), "Ethics in qualitative research and evaluation", *Journal of Social Work*, Vol. 3 No. 1, pp. 9-29, doi: [10.1177/1468017303003001002](https://doi.org/10.1177/1468017303003001002).
- Srnka, K.J. and Koeszegi, S.T. (2007), "From words to numbers: how to transform qualitative data into meaningful quantitative results", *Schmalenbach Business Review*, Vol. 59 No. 1, pp. 29-57, doi: [10.1007/BF03396741](https://doi.org/10.1007/BF03396741).
- Stager, G.S. (2016), "Seymour Papert (1928-2016)", *Nature*, Vol. 537, p. 308, doi: [10.1038/537308a](https://doi.org/10.1038/537308a).
- Stendal, K., Thapa, D. and Lanamäki, A. (2016), "Analyzing the concept of affordances in information systems", *2016 49th Hawaii International Conference on System Sciences (HICSS)*, IEEE, pp. 5270-5277, doi: [10.1109/HICSS.2016.651](https://doi.org/10.1109/HICSS.2016.651).

- Sun, L., Guo, Z. and Zhou, D. (2022), "Developing K-12 students' programming ability: a systematic literature review", *Education and Information Technologies*, Vol. 27, pp. 7059-7097, doi: [10.1007/s10639-022-10891-2](https://doi.org/10.1007/s10639-022-10891-2).
- Swidan, A., Hermans, F. and Smit, M. (2018), "Programming misconceptions for school students", *Proceedings of the 2018 ACM Conference on International Computing Education Research*, pp. 151-159, doi: [10.1145/3230977.3230995](https://doi.org/10.1145/3230977.3230995).
- Szabo, C., Sheard, J., Luxton-Reilly, A., Becker, B.A. and Ott, L. (2019), "Fifteen years of introductory programming in schools: a global overview of K-12 initiatives", *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, pp. 1-9, doi: [10.1145/3364510.3364513](https://doi.org/10.1145/3364510.3364513).
- Tiberius, V. (2019), "Scenarios in the strategy process: a framework of affordances and constraints", *European Journal of Futures Research*, Vol. 7, p. 7, doi: [10.1186/s40309-019-0160-5](https://doi.org/10.1186/s40309-019-0160-5).
- Tikva, C. and Tambouris, E. (2021), "Mapping computational thinking through programming in K-12 education: a conceptual model based on a systematic literature Review", *Computers and Education*, Vol. 162, 104083, doi: [10.1016/j.compedu.2020.104083](https://doi.org/10.1016/j.compedu.2020.104083).
- Tran, Y. (2018), "Computer programming effects in elementary: perceptions and career aspirations in STEM", *Technology, Knowledge and Learning*, Vol. 23 No. 2, pp. 273-299, doi: [10.1007/s10758-018-9358-z](https://doi.org/10.1007/s10758-018-9358-z).
- Ung, L.L., Labadin, J. and Mohamad, F.S. (2022), "Computational thinking for teachers: development of a localised E-learning system", *Computers and Education*, Vol. 177, 104379, doi: [10.1016/j.compedu.2021.104379](https://doi.org/10.1016/j.compedu.2021.104379).
- Viberg, O., Grönlund, Å. and Andersson, A. (2020), "Integrating digital technology in mathematics education: a Swedish case study", *Interactive Learning Environments*, pp. 1-12, doi: [10.1080/10494820.2020.1770801](https://doi.org/10.1080/10494820.2020.1770801).
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L. and Wilensky, U. (2016), "Defining computational thinking for mathematics and science classrooms", *Journal of Science Education and Technology*, Vol. 25 No. 1, pp. 127-147, doi: [10.1007/s10956-015-9581-5](https://doi.org/10.1007/s10956-015-9581-5).

**Corresponding author**

Niklas Humble can be contacted at: [niklas.humble@miun.se](mailto:niklas.humble@miun.se)