Ed Process Int J

www.edupij.com

ISSN 2147-0901 (Print), 2564-8020 (Online)

EDUCATIONAL PROCESS
INTERNATIONAL JOURNAL

Research Article

Cite this article: Govender, I., Govender, R. G., Govender, D. W. (2025). Robotics and Coding for Teacher Education: A Constructionist Approach. *Educational Process: International Journal, 15*, e2025176.
https://doi.org/10.22521/edupij.2025.15.176

Received February 2, 2025
Accepted April 19, 2025
Published Online April 24, 2025

Keywords: Computer programming, educational robotics, computational thinking, self-efficacy, coding

Author for correspondence:
Irene Govender
govenderi4@ukzn.ac.za
University of KwaZulu-Natal, South Africa

OPEN ACCESS

© The Author(s), 2025. This is an Open Access article, distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution, and reproduction, provided the original article is properly cited.

Robotics and Coding for Teacher Education: A Constructionist Approach

Irene Govender, Reginald G. Govender, Desmond Wesley Govender

Abstract

Background/purpose. This research explores the use of robotics to facilitate the learning of computer programming among non-specialist pre-service teachers with no prior programming experience. With the increasing demand for 21st-century teaching competencies, it is essential to equip future educators with computational thinking (CT) skills, even if they do not specialize in computer science. This study investigates the effectiveness of a constructionist approach using robotics in developing programming knowledge and CT skills among student teachers.

Materials/methods. The study was conducted at a university with 221 first-year pre-service teachers who were not majoring in computer science or related disciplines. A series of workshops were designed to engage participants in hands-on activities using robots, creating tangible learning outcomes in a simulated environment. A qualitative research approach was employed, utilizing reflective journals as the primary data source for analysis.

Results. The findings indicate that the constructionist approach, integrating robotics as a tangible learning tool, significantly enhanced participants' progressive understanding of programming constructs. Additionally, the intervention improved students' self-efficacy in learning programming and fostered the development of their CT skills.

Conclusion. The study demonstrates that robotics can be an effective tool for introducing computer programming to non-specialist pre-service teachers. The results contribute to the growing body of research on innovative pedagogical approaches for teaching programming and highlight the potential of robotics in developing 21st-century CT skills among future educators.

1. Introduction

South Africa faces a severe shortage of skills to meet the 4IR (Business Chief, 2020; Maisiri et al., 2021; ITWEB, 2022), especially computational thinking (CT) skills essential to solving complex problems in the 21st century. According to the literature, CT skills are developed most effectively through programming robots (Govender, 2022; Barth-Cohen et al., 2018) since logic is a fundamental component of CT. It is considered a basic competency for general problem-solving (Lee et al.,2023). As a result, the South African education department has acknowledged the value of coding and robotics education for students in schools. Hence, many schools have started to include coding and robotics education in their curriculum. To address the specific teaching needs of coding and robotics in the new curriculum, teachers with knowledge and training in programming are needed, which in turn requires that pre-service teachers be equipped with the necessary skills to meet the demands of the school curriculum. However, students who do not have formal programming training or are entering their first year of tertiary studies without previous programming experience (Lo et al., 2015; Saeli et al., 2011) find programming challenging.

Early exposure to the basics of coding is lacking in South Africa since coding and IT are only taught in the further education and training (FET) phase – the last three compulsory high school years. However, in other school specialization courses like mathematics, engineering, physical science, biology, etc., the material covered in earlier phases serves as scaffolding to study at the FET phase (Govender et al., 2021). Thus, without previous experience in programming, students find it challenging to build new knowledge from a deficient foundation, adding to the complexity of the subject.

Due to poor exposure to or experience of programming and, consequently, weak foundational knowledge of coding, many students tend to shy away from programs at tertiary levels that include computer programming. High dropout rates are also a result of the perception that computer programming courses are technical and difficult (Lin & Kuo, 2010). The study explores the effectiveness of using robotics to teach programming and develop computational skills among teacher trainees. The following questions have been formulated to guide this research.

RQ1: How do pre-service teachers experience programming using a simulated robotics environment?

RQ2: How does using robotics influence pre-service teachers' self-efficacy for learning programming?

The rest of the paper proceeds as follows: a brief overview of the literature and theory that underpins the study, including a discussion of the conceptual framework, is presented. Thereafter, the intervention is described, followed by the analysis, discussion, and conclusion.

2. Literature Review

There is consensus in the literature that learning to program is difficult (Govender, 2010, 2021). Teaching programming is also considered to be challenging due to the complexity and abstraction required to solve problems using computer programming languages (Bati et al., 2014; Olsson et al., 2015; Chen et al., 2017). Several methods have been used to teach programming to make learning simple. Some techniques used incorporate GUI environments, block-based coding, and visualization tools. However, these approaches have not been consistently successful. So far, there has been very little research on the most efficient ways to learn programming (Kalelioglu & Gulbahar, 2014; Techapalokul & Tilevich, 2017; Erol & Kurt, 2017). Hence, the issue of learning to program is still a concern, and hence, the development of CT skills remains.

Several studies have demonstrated that robotics has the potential to enhance the learning process in education (Chambers & Carbonaro, 2003; Jonassen, 2000; Kadeeva et al., 2020). Students can translate abstract concepts into real-world applications by experimenting with these technologies.

2.1. Constructivism and Constructionism in teaching and learning programming

Constructivism and constructionism are closely related learning theories, but their emphasis and application differ, especially in teaching and learning programming.

According to Papert's (1971) constructivist theory of learning, children learn through play and experimentation with technology. Therefore, the most effective way to learn is to be involved (Govender, 2022).

The constructivist approach promotes student-centered, discovery-based learning in which students apply what they already know to learn more. In contrast to lectures or step-by-step guidance, students gain knowledge through project-based learning, in which the teacher coaches them to connect different ideas and knowledge areas. For example, students learn programming concepts by experimenting with small code snippets and gradually constructing their mental models of programming.

Based on experiential learning and constructionist theories, it is believed that using robots can help enhance learning in the program challenge syndrome.

In addition, constructionism holds that people develop knowledge more effectively when they make tangible objects in real life (e.g., programs, robots, games). In this sense, constructionism is linked to experiential learning and builds upon Jean Piaget's (1945, 1957) constructivist theory but emphasizes learning through making and hands-on experiences. In programming education, this means students learn best by actively designing, coding, and debugging projects rather than just consuming theory. For example, a student learns programming by building and programming a robot, seeing immediate results, and refining their code based on trial and error.

Hence, it can be inferred that using robotics to learn programming and, hence, problem-solving is a constructionist method, making use of experiential learning. Another important aspect that affects students' behaviour in learning and performance is self-efficacy, which is discussed next.

2.2. Self-Efficacy and Programming

According to Hackett and Betz (1989), students' motivation and future academic preferences may be based on their perceptions of their competence.

Self-efficacy is the belief in being capable of achieving a specific goal or performing a specific task, in this case, coding and programming with robotics. Bandura (1994, p. 71) states that these beliefs "determine how people feel, think, motivate themselves and behave". According to the self-efficacy theory of motivation, four factors influence the self-efficacy of a particular task, which in turn influences behavior and performance. In the case of pre-service teachers learning coding and robotics, it is believed that increasing self-efficacy in learning to code will enhance their effectiveness in teaching it in schools. According to the literature, self-efficacy is a strong predictor of performance. In his seminal article, Bandura (1977) points out the effect self-efficacy has on performance and behavior.

Furthermore, it is now well-established from a variety of studies that self-efficacy is an important motivation for learning (Seon et al., 2019; Wang, 2023; Zimmerman, 2000). To this end, it can be inferred that students' computer self-efficacy can increase, thus influencing specifically female students' intentions to study computer-related courses (Govender & Khumalo, 2014). A study by Seturaman and Medley (2009) suggests that students' self-efficacy beliefs can be used to assess how well they are doing in their programming and their willingness to learn during difficult times. In another study, Jegede (2009) established that the novice learners' prior programming experience predicted Java programming self-efficacy. A very recent study (Daradoumis, 2022) affirmed the importance of self-efficacy in learning to program. We therefore explore another factor that could impact students' learning and performance: their self-efficacy to learn programming.

This study set out to explore pre-service teachers' learning experiences in learning coding with robotics with the view to not only improve the pedagogy of teaching programming but also to develop CT skills.

2.3. Conceptual framework

Using robotics to learn coding aligns with both self-efficacy and constructivism by fostering hands-on, student-centered learning experiences. Constructivism encourages learners to actively build knowledge through problem-solving and collaboration, while self-efficacy grows as students gain confidence in their ability to tackle programming challenges. Robotics provides an engaging, tangible way for students to experiment, make mistakes, and support each other, reinforcing their belief in their coding abilities and promoting collective self-efficacy through shared problem-solving.

Figure 1 illustrates Bandura's (1977) claim on the predictors of self-efficacy that affect behavior and performance.

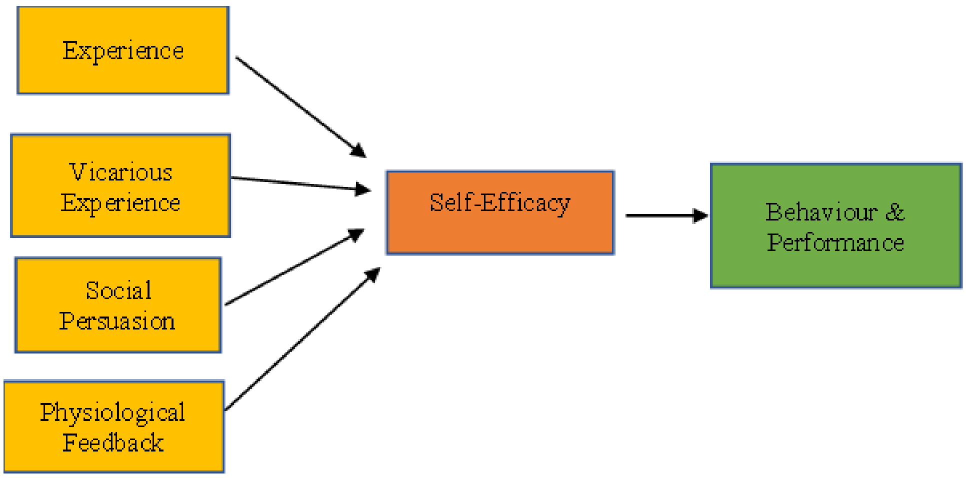

Figure 1. A Framework of self-efficacy theory of motivation (Bandura, 1977)

Experience refers to past experiences completing similar tasks. Having performed a similar task in the past, it is likely to be confident that the same task can be accomplished again in the future. It is most likely that the experience is most important. Hence, the idea of using robotics to make or create something tangible will elevate the experience, which links with the fourth factor, physiological feedback.

Vicarious experience - one can develop self-efficacy vicariously by watching other people perform the tasks. If one watches others similar to oneself (peers) perform a task successfully, one's self-efficacy can increase. The reverse is also true.

Social persuasion—Self-efficacy can increase if one receives encouragement when performing a task, gaining confidence. The same applies the other way around.

Physiological feedback -When confronted with a task, one experiences a sensation – how one interprets this feeling impacts one's self-efficacy. For example, the feeling of excitement or interest in learning a new aspect increases one's self-efficacy for that task or field of study.

The next section presents the details of the methodology followed.

3. Methodology

3.1. Study design

An intervention, the Design-Based Research (DBR) approach, was used. DBR involves three stages: planning, experimentation, and analysis (Gravemeijer & Cobb, 2006). The planning stage entails planning and designing the activities, the experimenting stage entails the accomplishment of the designed tasks by the students, and the analysis stage entails examining the actions and experiences of the students for further insights and improved learning. To determine students' experiences in using robotics and its effect on programming and problem-solving, a series of workshops were conducted using robotics to develop a prototype- i.e., to make something in a simulated environment.

3.2. Intervention

Since the course was focused on introducing first-year pre-service teachers who did not specialize in teaching computing or programming, Scratch, a commonly used visual programming language, was used in teaching coding as a prelude to robotics. We introduced students to the fundamentals of programming using Scratch, and thereafter, it was integrated with a robotic component for developing prototypes. Designing Scratch projects allows students to learn to think creatively, reason systematically, and work collaboratively, which are hallmarks of CT skills. Moreover, motivation, interest, and the belief that they can learn a subject (or field) are necessary for student success (Douglas et al., 2019; Blotnicky et al., 2018; Firat et al., 2018). Hence, students' self-efficacy was also explored in the study.

The teaching of coding and robotics was conducted over a period of four weeks. The four sessions entailed designing, coding, and testing prototypes using the microbit Makecode kit. The sessions included guided activities that were designed to build upon previous work.

The sessions were conducted online due to the effects of the COVID-19 pandemic, using a simulated environment, and a further face-to-face hands-on practical session was conducted to run their code on the actual physical microbit towards the end of the four-week session. Learning was facilitated online using Moodle LMS. Hence, the online discussion forum was used to communicate and resolve student queries. Figure 2 indicates the MakeCode microbit that was used.

Figure 2. MakeCode Microbit toolkit

3.3. Participants and Procedures

354 first-year pre-service teachers who were registered for a computer course as part of their teacher education curriculum, with no past experience in computer programming at the university, made up the population. However, only 221 students participated in the study in that they completed the reflective journal entries. This accounts for at least a 62% response rate. The demographic profile of the students is shown in Table 1.

https://doi.org/10.22521/edupij.2025.15.176 Published online by Universitepark Press

Table 1. Demographic Profile of Participants

Characteristics	Frequency	%
Gender: -		
Male	63	28.5%
Female	158	71.5
Ethic group: -		
African	215	97.2%
Indian	6	2.8%
Age group: - 20-25	156	70.5%
26-30	45	20.3%
31-35	14	6.4%
>36	6	2.8%
Qualification: -		
BEd- Foundation Phase	124	56%
BEd Intermediate Phase	97	44%

3.4. Data collection tools

Reflective journals were used to capture the experiences and perceptions of the pre-service teachers. Thorpe (2004) asserts that through reflective journaling, learners can gain a deeper understanding of their thinking patterns and engage with their instructors meaningfully to illuminate them. Farrah (2012, p.998) has demonstrated the value of reflective journaling, finding that it makes a "course enjoyable, motivating, and exciting."

Students were guided in writing general entries about their feelings, challenges, and how they solved problems. In addition to general entries, students were required to reflect and make specific entries related to each of the sessions and their activities. It has been shown that making learners aware of their programming abilities contributes to student success in computer programming courses (Kantathanawat et al., 2023).

3.5. Analysis

The journal entries were read several times to enhance understanding of pre-service teachers' coding experiences with robotics. Thematic analysis was used to analyse the journal entries., Journal entries were grouped and coded according to a priori themes based on the self-efficacy theory that determined their perceptions of learning coding through robotics. Gibbs (2010) refers to this type of coding as concept-driven coding. In addition to concept-driven coding, Gibbs (2010) advocates for open and axial coding to analyze the data further. In order to arrive at a set of holistic categories, both approaches to coding were used (Gibbs, 2010). The open coding process involved identifying and naming codes. More abstract categories were formed using axial coding based on the relationships among the different codes. Researchers in the project analysed the data triangularly, ensuring validity and researcher triangulation.

3.6. Ethical considerations

Since the journals were part of their submissions for assessment, they were informed that the selection of the journal entries would not infringe on their confidentiality, as journal entries were given pseudo-name identities to preserve anonymity. Informed consent was obtained from participants through their voluntary participation by completing the journal entries for submission for participation in the research and possible publication, of which they were informed at the beginning of the teaching course. The journal was part of the assessment and learning technique. However, only 221 (out of 354) were submitted for research.

4. Results

This section presents the analysis of the data obtained from the journal entries that described the experiences of first-year pre-service teachers learning coding and robotics in a simulated environment, with a view to determining students' self-efficacy to learn programming and suggesting possible implications of using the teaching strategy. Specifically, it reports on students' experiences with the strategy and the successes and challenges encountered. Supporting quotations from participants concerning the conceptual framework, including self-efficacy, are presented. To present a coherent discussion, first, each session with its subsequent activities, together with journal entry excerpts from the corresponding sessions, is presented, showing the learning progression. The thematic analysis and relevant excerpts then follow this to support the narrative.

4.1. Description of Workshop Sessions

Session One

The first introductory session helped participants become familiar with the user interface of the software environment, Makecode by Microsoft, and the BCC micro-bit components. Additionally, the history of computer programming and a brief introduction to coding were also presented. A step-by-step guided activity that introduced and applied a programming concept to a task was presented. Students were given tasks to complete, such as coding their names while scrolling on the screen and drawing a heart icon by pressing a button. There were 159 unique entries for this session. An excerpt from this session:

> *"It was a bit of a rocky start for me getting it up and running; I referred to the instructions time and time again to make sure I was doing every step correctly."*

Session Two

In this session, students were partially directed while they self-discovered the solution to programming tasks – hence, self-discovery formed the basis of this session. In this session, participants learned conditional statements (if—then), building on their prior knowledge and fundamental Scratch code setup (if statement). They learned about the forever loop and created a song. Participants were excited to make code and see the effects at this early stage. The session's high point was that they were willing to try further activities such as designing and coding. There were 185 journal entries for this session. An excerpt from this session:

> *"This program was very fun to code, as I got to try the new music blocks. I have to play with this a little more".*

Session Three

This session demonstrated animation with the repeat statement. Many students found some of the activities challenging but exciting. In this session, students were beginning to develop their practice skills and independently advance their learning. There were 145 journal entries for this session. An excerpt from this session:

> *Today's program was a bit challenging; I had to create an animation using the repeat statement. I have to say that I struggled a bit to make the loop work well, but I was able to make it work. At first, we had to add the repeat block and then add the icon after one another, which was wrong because you only add the repeat block once".*

Session Four

This session built upon the other sessions, where they could use the steps involved in computational thinking, including problem-solving. Using the microturtle (A LOGO-like turtle library for the microbit, which is available in MakeCode), they were required to draw a house specifically. Although some students struggled, they eventually realized the mistakes or misunderstandings, collaborated with peers and achieved their goals. This session and its subsequent activities involved more in-depth thinking, i.e., drawing using microturtle and reflection, involving decomposition, pattern recognition, abstraction, and algorithm, which are characteristics of CT. An excerpt from this session:

> *" We had to complete this activity and use our creativity as well as mathematical skills, the incorporation of numbers in the coding program proved to be a challenge for me, working through this activity very slowly and carefully, I started to realize that using the knowledge I already had and combining it with the new information actually helped me to solve the math's problems faster and more easily, I was astonished at this and quickly started trying different numbers and variables to see what the outcome would be".*

4.2. Thematic Analysis

The analysis is arranged according to the research questions formulated.

4.2.1. RQ1: Pre-service teachers' experiences in coding and robotics

The key themes that were identified in the general journal entries were excitement, challenge, difficulty, collaboration, eagerness to learn, and critical thinking. For each theme, a description and illustrative comments from students' reflective journals are presented below. Students learning to code with robotics experienced several challenges and excitement. What was most interesting and significant about their experience was their willingness to learn and zeal to conquer something new. Most students did not have previous exposure to programming. Using the microbit in designing and programming the tasks kept them engaged and motivated.

4.2.1 Excitement/Fun- Most students either described the activities and their actions as exciting and fun or enjoyed the learning experience at some point in the process. Despite having had some infrastructural challenges, they showed the intention to embrace coding and robotics. There were 78 entries that alluded to the word excitement, and 99 instances of the word fun in doing the activities. 115 entries indicated that they enjoyed the activities. The following excerpts below support this inference.

> *Fit example, this was an exciting code activity; my lecturer had guided us through the coding process and layout, which made it a bit easier to understand; however, I had to use a variable to change the number of steps and input c blocks to display the instruction.*

> *Introduction: It felt exciting to start a new section. I feel this section will be much more hands-on.*

> *It gets so exciting learning computer programming, challenging and enjoyable playing around with that computer.*

> *Making Compass code was really exciting, identifying those compass coordinates and setting our variable, which is "degrees." It was really exciting seeing my code running in MakeCode.*

> *Coding and Robotics test-…. There was nothing new, it included the codes and the blocks I am familiar with. The exciting part is that I have passed and excelled in it. Thus, this is the greatest achievement!*

The next theme identified was that learning to code with robotics was challenging.

4.2.2 Challenge - What is significant in this theme is that in most excerpts, the word "challenging" was coupled with fun or enjoyment. From the excerpts, it can be seen that despite the challenges experienced in developing the code to achieve the goal, it was still fun to do. There were 100 entries that alluded to the activities or learning as challenging and 87 entries that described their experiences as interesting or stimulating. This is in line with Paperts's theory of construction, i.e., to make something is to create excitement and enthusiasm to learn. While Papert's theory is related specifically to children, it applies to young adults as well. The following excerpts support this narrative.

> *However, coding makes me nervous, and sometimes, I feel as if I cannot go further because of how challenging it becomes. However, I am trying to work ahead of this fear and face my challenges head-on to succeed.*

> *It was just that it was my first time doing code that involved exponents, and I was not aware of how exponents are represented when coding. I then did some research on YouTube, and I found out that we use the Math block with two stars to show exponents. The code was amazing for me, and it was successful. I really enjoyed it!*

> *Robo act 3.2. It was challenging, but I asked my classmate for assistance, and I managed to finish the task.*

> *Coding building blocks is always challenging, not enjoyable at all. However, I could code a movement of the shark but could not change its costume to make it look real.*

> *Simulating a clock seemed challenging as I did not know where to find the 9 pictures*

> *I hardly understood how variables work in scratch programming. Learning using microbit is fun. you get to learn new skills during your programming period. I learned that variables can be changed anytime and you could name them whatever you like, even though I had challenges on successfully using some blocks like motion and sound. Coding building blocks was challenging but fun.*

The next theme is, as expected from the literature that programming is considered difficult by many first-time or novice learners of programming.

4.2.3 Difficulty - Many students found the tasks difficult; however, some assistance and the goal to succeed prompted them to persevere. There were 99 entries in which the tasks were described as difficult, and a further 67 entries described some tasks as hard. The list of excerpts below demonstrates the difficulties and determination to overcome.

> *I found out that some of them had difficulties with programming; they had little understanding of microbits*

> *The programming code is difficult, once you start thinking that you have it all under control, there is just that one program that will change everything that you believe you know and feel like a loser all over again.*

> *It was difficult for me to understand the activity, and I was nervous.*

> *I was so excited about today's lesson, but writing class activities was difficult because I got lost from the start. I did not understand the syntax of the language.*

> *This code was difficult to program; however, everything seems impossible until it is done, and nothing is too difficult. These coding programs are all working together to improve the coder's knowledge set starting now, and with a little practice, one can simply master the coding skill.*

4.2.4 Collaboration—It is necessary to develop collaborative skills, which are among the top 10 skills needed in the 4IR era. Most students referred to working with a peer or asking a friend for assistance when they got stuck on some aspect of the task. There were 86 entries that alluded to the idea of asking for assistance or help or collaborating (21 entries) with other members. Some excerpts to illustrate their willingness to collaborate are provided below.

> *Students also gain transferable life skills, design thinking, communication, and collaboration*

> *A variety of exercises taught me how to use a micro bit to create products or prototypes, including electrical, prototyping, testing, and coding. While learning about robotics, I also gained useful life skills, including collaboration, communication, and original thought. We can also learn skills such as teamwork and collaboration, problem-solving, and the ability to fail and try again. I am getting excited as we continue to learn about coding. It is becoming interesting.*

https://doi.org/10.22521/edupij.2025.15.176 Published online by Universitepark Press

To learn more about the blocks and how to program them on the computer, I continued to study some tutorials on my own and in collaboration.

For me to understand it, I had many questions, but my peers helped me out. I had trouble when writing a program for the microbit to simulate a clock watch. The blocks became too complicated, and I could not complete the work. Having done this programming, I was overwhelmed and was encouraged to be more collaborative with other students.

4.2.5 Eagerness to learn - It has been observed and inferred from the journal entries that students are resilient about learning programming, even though it is at a basic level. What has come through clearly is their interest and engagement in this aspect of the course. When they did acknowledge that they had missed previous lectures on which coding and robotics were based, they went back to review the material. 81 entries alluded to the idea that they were eager, keen, enthusiastic, willing, or happy to learn to code and know what needs to be done to master this aspect. The set of excerpts below illustrates their eagerness to learn.

All the coding challenges I have faced have made me eager to learn more about micro-bit coding. So I took my time and went online on YouTube and searched for microbit coding videos. The videos taught me that it may seem hard to make a program, but if you are keen on doing the program, it is not that hard.

This code allowed for some questioning about whether this program could vary. It was a bit tricky and complicated at first, but through some research and understanding, I was able to complete it.

By using new teaching materials, technical education and computer science can be made more meaningful and easier to understand

As I learned about animation, I became more excited as this included more fun things, such as pictures and puzzle-based blocks. Even my friends liked this chapter much more than me and I always tell myself that practice makes perfect.

This information was very resourceful to me. It made me eager to know if the children will be able to process this kind of information.

4.2.6 Critical/Computational thinking - As section two indicates, critical thinking is an important skill for the 21st century in the 4IR era. Learning to code robots allows one to develop critical thinking. 74 entries referred to their reflection, reasoning, and thinking patterns that indicated critical or computational thinking at play in the learning process or when solving the tasks. The excerpts below illustrate this inference.

I learned that learning coding as a child would help them strengthen their critical thinking skills and also enhance their ability to work in a team.

I think the micro bit gives me critical thinking and more creativity since it is inspirational.

In the last session, students were given tasks to develop critical computational thinking skills. The following excerpts support this narrative.

decomposition, pattern recognition, subtraction, and algorithms I think this is a great combination of math and computer, and I love that we can use the two together to create something amazing

This part was more successful since I also referred to problem 4. However, I failed to divide the total using the microbit. …. However, my challenges surfaced when I had to set the loop that would repeat the random number between 100. I create the basic structure, including the on-start and the repeat loop. However, I faced difficulties when it came to inserting the interest rate on the microbit as well as calculating the months that are required for the monthly R100 that is saved plus the interest to accumulate to 1 million.

I am writing for the last time in this journal. Capturing computer moments has been fun. This journal has become my friend for the last few weeks; it is so sad that I am about to say goodbye to my friend (XXX, this is the name I gave her). From this day onwards, I think I will have my own journal where I will write down all the moments that I would like to keep.

https://doi.org/10.22521/edupij.2025.15.176 Published online by Universitepark Press

4.2.2 RQ2: Pre-service teachers' self-efficacy for learning programming

Based on the conceptual framework depicted in Figure 1, the predictor constructs of self-efficacy (experience, vicarious experience, social/verbal experience, and physiological feedback/emotion) are explained with illustrative quotations.

Experience - As indicated earlier, one of the most important predictors of self-efficacy is experience. As the course progressed, it can be observed that even though most students had no prior experience with coding, they developed the exposure and experience. It shows that by the end of the course, students' experience increased their confidence in learning to program.

> *Excited to be doing a course with programming, as I have experimented in my private capacity with coding*

> *Having had this programming experience, I was overwhelmed and was encouraged to collaborate more with other students.*

> *Surprisingly, the steps were very easy and understandable. It was such an amazing experience of coding, and the instructions given to us students were very clear*

> *It was a great experience to touch the micro: bit kit, USB cable, battery holder, 1x set of 10 crocodile cables, microbit V2, etc.*

> *Thus, it was a great experience as I even learned how to compose a song in any way that I wanted it to be.*

> *Working through this activity very slowly and carefully, I started to realize that using the knowledge I already had and combining it with the new information actually helped me to solve the math problems faster and more easily, I was astonished at this and quickly started trying different numbers and variables to see what the outcome would be.*

> *Using building blocks in programming requires a great deal of critical thought and excitement. I completed all of the follow-up exercises, acquired experience, and faced challenges that helped me comprehend every concept and construct the necessary building blocks to accomplish everything.*

> *I am using this understanding to assist my peers who have less experience and, in so doing, find myself learning more in the process*

Vicarious Experience - Another predictor of self-efficacy is watching other people perform the tasks. If one watches others similar to oneself (peers) perform a task successfully, one's self-efficacy can increase. In this course, it has been observed that students saw their peers learn this course and perform tasks successfully.

> *I tried looking for tutorials that were similar to these tasks but I could not find any. Planning on discussing this activity with a peer [on] how they said they understand it.*

> *I understood it so well that I was able to explain it to a friend of mine.*

> *I ended up calling a group member for some clarification on what a micro bit is and how it functions because I did not attend the class during the week. I started practicing how to make the loop work, and I really failed until a friend advised me to start by using a block sign first.*

It is also true that watching or being with peers who do not achieve success in the learning activities can have a negative effect on one's self-efficacy, as the next excerpt indicates.

> *The inclusion of models and equations just makes things worse for me to the point I do not know what to understand these activities. I tried getting help from my peers and my tutor, but I do not see any progress.*

Social/verbal persuasion - The third construct, verbal persuasion, of self-efficacy, helps one to learn further or to persevere when one is encouraged. This encouragement can come from peers, parents, and teachers/instructors.

> *I came across some interesting news that micro: bit was designed to encourage children to get actively involved in writing software for computers and building new things, rather than being consumers of media.*

A little tricky especially when it came to variables, but I am glad Mr [….] was there to help as during this time I had been going through personal difficulties at home with power outages for 3 weeks, I made arrangements to perform my tasks on time and listen to recorded lectures where possible.

I however felt programming the mirco: bit to send out an SOS, using specific conventions, were long and tedious. However, Prof […] suggested using a looping structure, when creating this SOS program, which is an interesting idea, that could actually work.

I am a learner who enjoys theory more than practical however for this session I am enjoying playing around with the program and discovering new things as we go along.

Physiological feedback/emotion—The fourth construct, Physiological feedback, occurs when confronted with a task. One experiences a sensation that may be perceived as positive or negative. This perceived feedback interpretation, whether positive or negative, can affect one's self-efficacy. If perceived or interpreted as positive, it would positively affect one's self-efficacy for learning programming.

For this construct, the excerpts to support the narrative come from the themes identified in the first section of the thematic analysis: How do pre-service teachers experience learning to code robots? Some excerpts that show how students interpret the sensation when confronted with the tasks given to them are presented below.

The robotics have not been as fun as I expected them to be. Maybe online learning affects that. Despite all this, I am still looking forward to creating more codes and programming.

I have been trying to check my previous codes using trace tables. I like the idea because they are one of the accurate ways of defining the meaning of codes. It is not yet that easy, but I do believe that as I keep on practicing, I will get used to it.

Powers—At first, I was hesitant about whether I understood the question, which almost led to my giving up. However, I read it over and over again until I succeeded, and my code was just right!

At first, I was truly confused about the program and how to navigate to get the correct block. I was guided through the follow-up exercises and surprisingly found the steps easy and understandable. It was such an amazing experience of coding.

5. Discussion

This study explored pre-service teachers' experiences using robotics and the role self-efficacy plays in learning programming. The main source of data came from the reflective journals. Interestingly, an important finding emerged from this study, affirming Farrah's (2012) finding that reflective journaling made the "course enjoyable, motivating and exciting ."An excerpt from the journal evidences this. "Capturing computer moments has been fun. This journal has become my friend for the last few weeks; it is so sad that I am about to say goodbye to my friend (XXXX, this is the name I gave her)". This, in turn, helped students further understand the content by constantly reflecting on their learning and practice.

Concerning the first research question, the first four themes (excitement, challenge, difficulty, collaboration) described the participants' experience of learning coding and robotics. Interestingly, most students indicated that aspects of the course were challenging or difficult but overcame the challenge by the excitement and fun experienced in learning to make something. Being engaged and involved in making something tangible played a crucial role in students' learning and the intention to learn further. These findings align with what Kadeeva et al. (2020, p. 7) claim in their study that robotics assists in a better understanding of the material due to the "meta-subject connections of robotics with other school subjects ."The excitement, challenges, difficulties, and collaboration experienced during this course have contributed to students' self-efficacy. These experiences may be linked to the experience construct as it relates to the self-efficacy construct in Figure 1. It may be inferred that the themes of excitement, difficulty, challenge, and collaboration can be considered part of the experience construct.

https://doi.org/10.22521/edupij.2025.15.176 Published online by Universitepark Press

In addressing the second research question, the strategy of learning to program using robotics is a successful way to bring students/learners on board to learn programming. Most students were in the age range between 20 and 25. It is a commonly held belief that young students in this age range can more readily learn technical skills. Hence, by increasing students' self-efficacy to learn programming, they are eager to advance their learning and consequently develop critical thinking and CT skills. The constructionist approach contributed to their excitement, challenges, difficulties, and collaborative skills, contributing to their positive experience. The themes, eagerness to learn, and critical/computational thinking are effects of their engagement and motivation. These findings agree with both Seon et al. (2019) and Wang's (2023) research, which showed that self-efficacy is an important motivation for learning. This is an important aspect, particularly for females who have been generally slow to take up programming, as indicated in Govender and Khumalo's (2014) study. In this study, most participants were females (see Table 1), which showed the impact of learning in this way.

For this reason, we believe that the conceptual framework (Figure 1) can be modified for this learning experience of learning to program using robotics, as depicted in Figure 3.

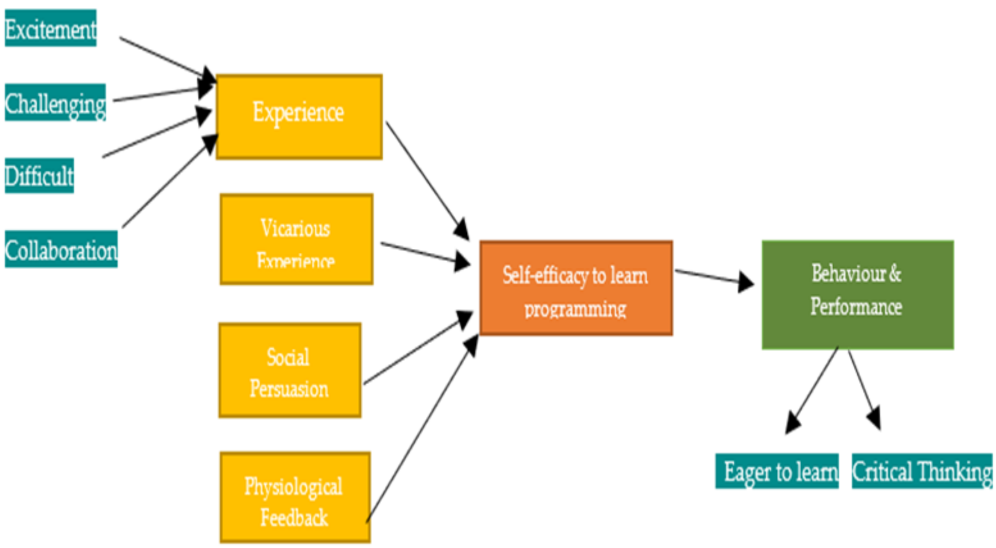

Figure 3. Modified conceptual framework for learning coding and robotics self-efficacy

6. Conclusion

The study set out to explore pre-service teachers' learning experience with coding and robotics and, hence, to inform the teaching strategy that is effective for first-time learners of programming. Using the framework of self-efficacy theory of motivation (Figure 1) allowed the authors to highlight the four constructs/themes (Excitement, Difficulty, Challenging, and Collaboration) that fit into the Experience construct of the framework. This allowed for an appropriate response to the construct Behaviour and Performance in that students were Eager to Learn and develop Critical Thinking skills, resulting in a modified conceptual framework for learning to program, as illustrated in Figure 3. The results have shown that using robotics to introduce computer programming is an effective approach. The idea of increasing interest and the eagerness to learn coding has been established using a constructionist approach. Most students who participated in the reflective journal entries showed extreme interest in continuing to learn to code.

The evidence from this study suggests that using robotics (making or constructing something) in teaching programming has proven to be effective in increasing their self-efficacy to program. Some implications for teaching include engaging students in learning activities that have proven to be exciting and stimulating. One such activity is using robotics and coding to enhance self-efficacy in learning to code.

https://doi.org/10.22521/edupij.2025.15.176 Published online by Universitepark Press

The current study contributes to the learning-to-program literature in three important ways. First, it highlights the success of incorporating robotics in teaching programming. Second, the study demonstrates that using robotics can enhance self-efficacy when it comes to learning to program. Thirdly, the study highlighted a deeper understanding of reflective journals to inform teaching practices in programming.

7. Limitations

The study was conducted within the specific context of a university course with bachelor of education (BEd) students at the university. This specific educational setting, with its student demographic, may restrict the generalisability to other contexts. Additionally, the study used a convenience sample, so there might be potential biases. Despite its exploratory nature, this offers insight into the pedagogical approaches to introducing programming to enhance 21st-century skills.

Declarations

Author Contributions. I.G.: Literature review, conceptualization, data analysis, original manuscript preparation. R.G.: Intervention, data collection, methodology. D.W.G.: review-editing and writing. All authors have read and approved the final version of the article.

Conflicts of Interest. The authors declare no conflict of interest.

Funding. This work was supported by the grant awarded to R.G. by the National Research Foundation (NRF) in SA [grant number TTK2204092902].

Ethical Approval. Ethics Committee (HSSREC) of the University of KwaZulu-Natal (Protocol reference number: HSSREC/00006672/2024

Data Availability Statement. The data sets analyzed during the current study are not publicly available due to confidentiality considerations but are available from the author(S) upon reasonable request.

Acknowledgments. Not applicable

References

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, *84(2),* 191–215. https://doi.org/10.1037/0033-295X.84.2.191

Bandura, A. (1994). Self-efficacy. In V. Ramachaudran (Ed.), Encyclopedia of human Behaviour, *(4),*71-81. New York: Academic Press.

Barth-Cohen, L.A., Jiang, S., Shen, J., Chen, G., & Eltoukhy, M. (2018). Interpreting and navigating multiple representations for computational thinking in a robotics programming environment. *Journal of STEM Education Research*, *1(1),* 119-147. https://doi.org/10.1007/s41979-018-0006-2

Bati, T. B., Gelderblom, H., & Van Biljon, J. (2014). A blended learning approach for teaching computer programming: design for large classes in Sub-Saharan Africa. Computer Science Education, 24(1), 71-99. https://doi.org/10.1080/08993408.2014.897850

Blotnicky, K. A., Franz-Odendaal, T., French, F., & Joy, P. (2018). A study of the correlation between STEM career knowledge, mathematics self-efficacy, career interests, and career activities on the likelihood of pursuing a STEM career among middle school students. International *Journal of STEM Education, 5(1)*, 1-15. https://doi.org/10.1186/s40594-018-0118-3

Business Chief. Available online: https://businesschief.eu/leadership-and-strategy/digital-skills-south-african-graduates-will-need-2020 (accessed on 12 June 2022).

Chambers, J.M., & Carbonaro, M. (2003). Designing, Developing, and Implementing a Course on LEGO Robotics for Technology Teacher Education. *The Journal of Technology and Teacher Education,* *11*, 209–241. Retrieved March 16, 2025, from https://www.learntechlib.org/primary/p/14607/.

Chen, Q., Tang, Y., Li, L., Yang, G., Yang, M., Xie, Z., … Huang, R. (2017). A practice on Lego Mindstorms for Computer Science Freshman experimental education. *Destech Transactions on Social Science, Education and Human Science, 1*, 17-21. DOI 10.12783/dtssehs/aeme2016/7454

Cohen, L., Manion, L., & Morrison, K. (2017). Research methods in education, (8th ed.). Routledge. https://doi.org/10.4324/9781315456539

Daradoumis, T., Marquès Puig, J.M., Arguedas, M. et al. Enhancing students' beliefs regarding programming self-efficacy and the intrinsic value of an online distributed Programming Environment. *J Comput High Educ*, *34*, 577–607 (2022). https://doi.org/10.1007/s12528-022-09310-9

Douglas, M. E., Peecksen, S., Rogers, J., & Simmons, M. (2019). College students' motivation and confidence for ePortfolio use. *International Journal of ePortfolio, 9(1*), 1-16. http://www.theijep.com

Erol, O., & Kurt, A. A. (2017). The effects of teaching programming with scratch on pre-service information technology teachers' motivation and achievement*. Computers in Human Behavior, 77*, 11–18. https://doi.org/10.1016/j.chb.2017.08.017

Farrah, M. (2012). 26(4), 2012Reflective Journal Writing as an Effective Technique in the Writing Process. *An-Najah University Journal for Research - B (Humanities), 26(4)*. DOI: 10.35552/0247-026-004-008

Fırat, M., Kılınç, H., & Yüzer, T. V. (2018). Level of intrinsic motivation of distance education students in e-learning environments. *Journal of Computer Assisted Learning, 34(1),* 63-70. https://doi.org/10.1111/jcal.12214

Gibbs, G. (2010). Analyzing Qualitative Data. Los Angeles: Sage.

Govender, I. (2022). Research Status in Computational Thinking in STEM Education. In M.

Kalogiannakis & M. Ampartzaki (Eds), Advances in Research in STEM Education, IntechOpen Publishers.

Govender, I. & Khumalo, S. (2014). Reasoned Action Analysis Theory as a Vehicle to Explore Female Students' Intention to Major in Information Systems*. Journal of Communication, 5(1)*, 35-44.

Govender, I. (2021). Towards understanding information systems students' experience of learning introductory programming: A phenomenographic approach. *Journal of Information Technology Education: Innovations in Practice, 20,* 81-92. https://doi.org/10.28945/4782

Govender, I. (2010). From Procedural to Object-Oriented Programming (OOP) – Performance n OOP: An empirical study. *South African Computer Journal (SACJ), 46*, 14-23.

Govender, R. G., & Govender, D. W. (2021). A Physical Computing Approach to the Introduction of Computer Programming among a Group of Pre-service Teachers. *African Journal of Research in Mathematics, Science and Technology Education, 25(1)*, 91-102. https://doi.org/10.1080/18117295.2021.1924440

Gravemeijer, K., & Cobb, P. (2006). Design research from a learning design perspective. In J. Van den Akker, K. Gravemeijer, S. McKenney & N. Nieveen (Eds), Educational design research, Routledge: London, United Kingdom, p. 17-51.

Hackett, G.& Betz, N.E. (1989). An exploration of the mathematics self-efficacy/mathematics performance correspondence. *Journal for Research in Mathematics Education 20*, 261-273.

ITWEB. Available online: https://www.itweb.co.za/content/4r1ly7RblGE7pmda (accessed on 21 June 2022).

Jegede, P.O. (2009). Predictors of Java Programming Self-Efficacy among Engineering Students in a Nigerian University. *International Journal of Computer Science and Information Security, 4(1 & 2)*, 7.

Jonassen, D. H. (2000). Toward a design theory of problem solving. *Educational Technology Research and Development, 48(4)*, 63–85. https://doi.org/10.1007/BF02300500

Kadeeva, O., Belov, A., Syritsyna, V., Maksimenko, S., & Sinko, V. (2020). Methodological aspects of the implementation of additional education in robotics. E3S Web of Conferences 210, 22040, ITSE-2020. https://doi.org/10.1051/e3sconf/202021022040

Kantathanawat, T., Chinchua, S., & Tuntiwongwanich, S. (2023). Programming Self-efficacy in Higher Education Research: A Systematic Literature Review. *Journal of Higher Education Theory and Practice, 23(19),* DOI: 10.33423/jhetp.v23i19.6673

Lee, S.W.Y., Tu, H.Y., Chen, G.L. et al. (2023). Exploring the multifaceted roles of Mathematics learning in predicting students' computational thinking competency. *IJ STEM Ed, 10(64*). https://doi.org/10.1186/s40594-023-00455-2

Lin, H. T., & Kuo, T. H. (2010). Teaching programming technique with edutainment robot construction. In Proceedings of the 2nd International Conference on Education Technology and Computer, Shanghai, China, Date of Conference, 22-24 June 2010. https://doi.org/10.1109/ICETC.2010.5529557

Lo, C. A., Lin, Y. T., & Wu, C. C. (2015). Which programming language should students learn first? A comparison of Java and Python. In Proceedings 2015 International Conference on Learning and Teaching in Computing and Engineering LaTiCE, Taipei, Taiwan, 9-12 April 2015.

Maisiri, W., & Van Dyk, L. (2021). Industry 4.0 skills: a perspective of the South African manufacturing industry. *SA Journal of Human Resource Management, 19*, a1416. https://doi.org/10.4102/sajhrm.v19i0.1416

Olsson, M., Mozelius, P., & Collin, J. (2015). Visualisation and gamification of e-learning and programming education. *Electronic Journal of e-Learning, 13*, 441-454.

Papert S. Teaching children thinking (1971). *Contemporary Issues in Technology and Teacher Education, 5(3/4*), 353-365.

Piaget, J. (1945). Play, dreams, and imitation in childhood. London: Heinemann.

Piaget, J. (1957). Construction of reality in the child. London. Routledge & Kegan Paul.

Saeli, M., Perrenet, J., Jochems, W.M.G., & Zwaneveld, B. (2011). Teaching programming in secondary school: a pedagogical content knowledge perspective. *Informatics in Education, 10,* 73-88. https://doi.org/10.15388/infedu.2011.06

Seon, A. H., Bong, M., Renninger, K.A., Hidi SE. (2019). Self-Efficacy in Learning: Past, Present, and Future. In: The Cambridge Handbook of Motivation and Learning. Cambridge Handbooks in Psychology. Cambridge University Press, 63-86.

Seturaman, S. & Medley, M.D. (2009). Age and Self-efficacy in Programming. *Journal of Computer Sciences in Colleges 25(2*), 122-128.

Techapalokul, P., & Tilevich, E. (2017). Understanding recurring quality problems and their impact on code sharing in block-based software. In Proceedings 2017 IEEE Symposium on Visual

Languages and Human-Centric Computing (VL/HCC), North Carolina, United States of America, 11-14 October 2017. https://doi.org/10.1109/VLHCC.2017.810344

Thorpe, K. (2004). Reflective learning journals: from concept to practice. *Reflective Practice. 5(3),* 327-343, DOI: 10.1080/1462394042000270655

Wang, C.J. (2023). Learning and Academic Self-efficacy in Self-regulated Learning: Validation Study with the BOPPPS Model and IRS Methods. *Asia-Pacific Edu Res 32*, 37–51. https://doi.org/10.1007/s40299-021-00630-5

Zimmerman, B.J. (2000). Self-Efficacy: An Essential Motive to Learn. *Contemporary Educational Psychology 25*, 82-91.https://doi.org/10.1006/ceps.1999.1016

About the Contributor(s)

Irene Govender, PhD, is a Professor of Information Systems and Technology at the University of KwaZulu-Natal (UKZN). Her research interests are computer programming, educational technologies, and teacher education in computing and Big Data Analytics.
Email: govenderi4@ukzn.ac.za
ORCID: https://orcid.org/0000-0002-4499-1091

Reginald G. Govender, PhD, is a senior lecturer in the Department of Computer Science Education at UKZN.
Email: govenderR4@ukzn.ac.za
ORCID: https://orcid.org/0000-0002-3143-4050

Desmond W. Govender, PhD, is a Professor of Computer Science Education at UKZN.
Email: govenderd50@ukzn.ac.za
ORCID: https://orcid.org/0000-0002-6115-9635

Publisher's Note: *The opinions, statements, and data presented in all publications are solely those of the individual author(s) and contributors and do not reflect the views of Universitepark, EDUPIJ, and/or the editor(s). Universitepark, the journal, and/or the editor(s) accept no responsibility for any harm or damage to persons or property arising from the use of ideas, methods, instructions, or products mentioned in the content.*