Park, H., Boz, T., Sawyer, A., & Willingham, J. (2026). Exploring robot programming in a geometry content course: Learning opportunities for prospective teachers. *Contemporary Issues in Technology and Teacher Education, 26*(1), 85-115.

Exploring Robot Programming in a Geometry Content Course: Learning Opportunities for Prospective Teachers

Hyejin Park
Drake University

Tuğba Boz
University of North Dakota

Amanda Sawyer & James C. Willingham
James Madison University

This study evaluated programming and robotics (PR)-integrated geometric learning activities designed to build prospective teachers' (PSTs') knowledge and skills necessary for incorporating PR in elementary geometry classrooms. To identify the learning opportunities these activities provided to PSTs, the authors examined arguments PSTs generated to justify the correctness of programs designed for robots to travel along triangular paths, as well as their initial and postsurvey responses and written reflections describing their beliefs about learning and teaching mathematics with PR and demonstrating their learning experience through the PR-integrated activities. Data analysis showed three different domains of learning opportunities offered to PSTs for their knowledge development for teaching PR in mathematics classrooms: (a) developing an understanding of geometric concepts used in program design, (b) improving justifying skills of using geometric reasoning to verify the correctness of robot programs, and (c) building productive views toward learning and teaching mathematics with PR. The researchers also identified the specific knowledge and skills PSTs used to verify program correctness before testing with physical robots. Suggestions are proposed for teacher education to prepare PSTs for PR-integrated mathematics instruction.

Postpandemic classrooms use more technology than ever before (Tsakeni, 2022). The rapid shift to digital learning environments during the pandemic accelerated the integration of digital tools in classrooms. Such a change emphasizes that teachers must possess the knowledge and skills that help students use various technological tools in their learning. Mathematics teachers must diligently build this knowledge, as it enables them to leverage new technologies in teaching, making mathematical concepts more accessible and tangible to their students. In many countries, current recommendations and policy statements have evolved to encourage the use of programming and robotics (PR) in mathematics classrooms as pedagogical tools, not only to assist students in deepening their understanding of mathematics but also to provide computer science (CS) learning opportunities (e.g., Association of Mathematics Teacher Educators [AMTE], 2022; Bråting & Kilhamn, 2022; Department of Education, 2021; Grover & Pea, 2013; Kaufmann et al., 2023; Swedish National Agency for Education, 2020).

Teachers at every grade level are expected to integrate programming during their mathematics instruction. Thus, teachers must possess a subset of combined mathematics and programming knowledge for integrated teaching. However, many educators lack sufficient programming knowledge due to limited programming experience during their K-12 education and subsequent teacher education programs (Kim et al., 2022; Vegas & Fowler, 2020). This lack of experience poses a significant challenge for meeting new or evolving mathematics standards and for integrating CS concepts and applications into mathematics classrooms. Furthermore, the rapid evolution of technology, such as Artificial Intelligence-based tools (e.g., ChatGPT), exacerbates this issue by making it even harder for teachers to keep up without sufficient ongoing professional development and support (Sawyer, 2024).

It is imperative to address these gaps and needs in teacher education programs. During mathematics education coursework, mathematics teacher educators should provide learning opportunities for prospective teachers (PSTs) to build the knowledge and skills necessary for PR integration into math lessons (Manley & Park, 2024; Park & Manley, 2024). We define learning opportunities as “circumstances that allow students to engage in and spend time on academic tasks…” (National Research Council, 2001, p. 333).

To achieve this objective, as part of our research project, we created PR-integrated geometry learning activities for PSTs that engage them in designing block-based programs focusing on reasoning to move physical robots on specific paths of shapes, justifying the correctness of the programs before testing the programs with physical robots, running physical robots to test the programs, and revising the programs until robots move as they expected. We examined the effectiveness of PR-integrated activities for teacher learning by implementing them across three sections of a geometry content course for K-8 PSTs.

This article describes the details of the PR-integrated geometry learning activities used in the geometry content course to support PSTs’ learning. We report the learning opportunities these activities provided to PSTs, as evidenced by their work. We also illustrate the specific knowledge and skills PSTs used during the activities.

The data for our investigation was their feedback on learning from the PR-integrated activities and the arguments they constructed to demonstrate the correctness of the programs. The following research questions guided our study for this paper:

1. What knowledge and skills do PSTs use when asked to construct arguments justifying the correctness of programs designed for a robot to travel triangle paths?

2. What learning opportunities do the PR-integrated geometry learning activities provide PSTs based on their evaluations of these activities and their learning?

Related Literature

Use of Programming and Robotics in Learning and Teaching Mathematics

Integrating PR in mathematics classrooms can offer unique ways to teach mathematics (Humble, 2023) and increase students' interest in science, technology, engineering, and mathematics (STEM) (Meschede et al., 2022). Using programming in mathematics is not a recent endeavor. In the 1970s, Seymour Papert, a mathematician and computer scientist at the Massachusetts Institute of Technology, introduced the programming language Logo to support children's mathematical thinking. Papert was deeply interested in understanding how the mind works, and he sought to bridge the gap between artificial intelligence and human cognition by exploring connections between computer science and human cognition (Mitchell, 2019). This curiosity even led him to work with Jean Piaget in Geneva, where he studied Piaget's theories of cognitive development and constructivist learning (Ackerman, 2001).

These experiences profoundly influenced Papert and laid the foundation for his theory, *constructionism*. Constructionism claims that learners construct knowledge most effectively when they are actively engaged in making meaningful artifacts (Papert, 1980). Out of this vision came Turtle Geometry, a pedagogical approach that allowed students to program a virtual "turtle" to move across a screen and create geometric patterns, thereby learning mathematics through exploration and visualization. Throughout the 1980s and 1990s, researchers further explored the potential of using programming in math education (e.g., Abelson & DiSessa, 1980; Hoyles et al., 1986), with studies demonstrating that such tools could support spatial reasoning, problem-solving, and higher order mathematical thinking. However, due to limited access to computers, teachers did not widely adopt these innovations at that time (Boz & Allexsaht-Snider, 2022).

The landscape began to shift in 2006 when Jeannette Wing introduced the concept of *computational thinking* as a foundational literacy alongside reading, writing, and arithmetic. Since then, interest in PR in K-12 education has increased, primarily through the use of tangible educational robotics kits (e.g., Hickmott et al., 2018). Unlike earlier approaches centered on screen-based agents, today's technologies enable students to interact with physical robots, allowing them to see and manipulate the outcomes of their code in real-time. This physical embodiment of computation has opened new avenues for engaging students in mathematics. As Dennis and Buchbinder (2023) explained, these tools can evolve from external artifacts into instruments that shape and support mathematical thinking through a process known as *instrumental genesis* (Trouche, 2005).

The current literature highlights that teaching mathematics with PR can enhance elementary students' learning and increase their motivation and engagement in mathematics classrooms (e.g., Harper et al., 2021; Hickmott et al., 2018; Israel & Lash, 2020; Lopez-Caudana et al., 2020). It also encourages students to develop positive attitudes toward learning mathematics (e.g., Ke, 2014). It helps them connect their mathematics learning to everyday life experiences (e.g., Lambic, 2011), as PR provides an application outlet for mathematics, enabling students to see the usefulness of what they are learning.

Moreover, research has demonstrated that making abstract concepts more tangible through PR can deepen students' understanding of mathematics and serve as both a language and a manipulative tool for mathematics learning (Goldenberg et al., 2021). The tangible nature of robots also allows students to visualize the outcomes of their code. Therefore, educational robotics is increasingly recognized not only as a powerful tool for K-12 learners but also as a critical component in preparing teachers to teach coding and computational thinking, particularly as robotics continues to be a growing focus in the broader discipline of computer science and its associated industries, especially those connected to artificial intelligence (Zhang & Lu, 2021).

Existing studies in the literature offer different ways to interact with and conceptualize the core ideas of mathematics using PR (e.g., Suters & Suters, 2020). Specifically, many studies focus on geometry due to its natural connection to robotics (Forsström & Kaufmann, 2018; Manley & Park, 2024). This connection arises because many fundamental geometric concepts are applicable to the control of educational robots.

For example, programming a robot to move along a specific path often requires understanding coordinate systems, angles, and rotations. Empirical evidence shows that using PR in mathematics lessons can improve learners' understanding of geometric concepts.

In their review of prior studies on geometry learning through PR, Clements and Meredith (1992) highlighted that robot programming can help students develop "higher levels of geometric thinking" (p. 2), including an understanding of angle and length measurements. It also provides students with opportunities to analyze the properties of a shape and to construct an abstract definition of it. Results from Terwilliger et al.'s (2019) study also demonstrated how robot programming activities can even support teachers' understanding of generalization and geometry.

The literature, however, points out that the mere inclusion of PR in mathematics classrooms does not enhance mathematical learning. Its effectiveness depends on how it is integrated into mathematics classrooms, particularly in relation to the tasks or pedagogical approaches teachers use to assist students' learning of mathematics through programming (e.g., Zhong & Xia, 2020).

For teachers, PR-integrated mathematics teaching can be challenging (Holo et al., 2022; Papadakis et al., 2021). Elementary teachers often feel unprepared for such integration instruction for various reasons, such as due to lack of training on the use of PR and lack of the pedagogical and content knowledge required for integrating PR in mathematics classrooms, though they see positive effects on teaching that way, such as making students' mathematics learning more enjoyable and more making sense (Humble et al., 2020; Vinnervik, 2022).

Furthermore, although many studies on professional development offering PR learning opportunities for practicing teachers acknowledge the organic relationship between STEM subject areas and PR (Kim et al., 2022; Yadav et al., 2022), there is limited research on how PSTs develop this understanding. Savard and Highfield's (2015) study found that elementary school teachers had a somewhat superficial understanding of the mathematical learning opportunities PR-integrated activities can offer, even after implementing these activities in their classrooms. This suggests a need for more targeted efforts in teacher preparation programs. These efforts should help future teachers recognize and leverage the connections between PR and mathematics learning and should offer them opportunities to reflect on their mathematics learning with PR from both a learner's and a teacher's perspective.

Silva et al. (2024) highlighted the importance of monitoring PSTs' perceptions of PR use, as these perceptions shape PSTs' willingness to integrate PR into their instruction. Our study targeted prospective elementary teachers enrolled in a teacher education program. It aimed to help them develop the knowledge and skills necessary to integrate PR into the mathematics classroom and to build positive perspectives on learning and teaching mathematics with PR.

Proof in Mathematics and Computer Science: Why Justification Matters

Being able to justify a claim with valid reasons is an essential skill in both proving and programming activities. The National Council of Teachers of Mathematics (NCTM) Process Standards (2000) stated that during the journey of learning school mathematics, students across all grade levels should be able to "recognize reasoning and proof as fundamental aspects of mathematics" and "develop and evaluate mathematical arguments and proofs" using various types of reasoning and methods of proof (p. 56). Aligned with NCTM's recommendations, the Standards for Mathematical Practices also illustrated that, while learning mathematics, proficient students are expected to construct arguments using deductive reasoning and to critique the reasoning of others (National Governors Association Center for Best Practices & Council of Chief State School Officers, 2010).

Knuth (2002) highlighted the explanatory role of proof in demonstrating the truth of a statement and advocated using proofs in school mathematics as a tool to enhance students' understanding, as they can help them make sense of mathematics. Comparing proofs that prove and proofs that explain, Hanna (1989) also asserted that mathematics educators "should give a more prominent place in the mathematics curriculum to proofs that explain" (p. 54) because such a proof provides "a set of reasons that derive from the phenomenon itself" (Hanna, 1990, p. 9).

Therefore, having students focus on understanding the reasons provided in explanatory proofs can help them grasp key mathematical ideas and enhance their ability to discern the truth or falsity of mathematical statements. In the context of a mathematics classroom, Dogan and Williams-Pierce (2019) described three phases of proving activities that students can engage in, involving phases of exploration for pattern identification and generalization for making a conjecture (or claim), justification of the conjecture (claim) informally or formally, and evaluation of one or more justifications by a (classroom) community as to whether the mathematically argument (or arguments) is valid and acceptable. Stylianides (2008) also captured these activities as pivotal instructional components of proving activities. However, Bieda et al. (2013) pointed out that

> the curriculum used in elementary mathematics classrooms may not be sufficient, as written, to provide students with meaningful opportunities to learn to generate and evaluate mathematical claims; the average percentage of tasks we sampled across seven elementary mathematics textbooks related to RP [reasoning-and-proving] was only 3.7%. (p. 9)

Thus, the role of a teacher is essential in increasing students' opportunities to learn to make and justify mathematical claims using accepted statements (e.g., definitions and theorems) and mathematical forms of reasoning and representations.

Similar to the verification role of proof in mathematics (de Villiers, 1990), proving a computer program is crucial in CS for verifying its correctness without doubt about the existence of program errors, thereby "increasing confidence in the correctness of running programs" and "providing additional understanding of why the programs work as they do" (London, 1970, p. 281). In CS, proving programs is not a common practice, however. From a practical standpoint, this makes sense. The literature on programming reveals that novice programmers often employ a "trial-and-error" strategy (Merisio et al., 2021, p. 191) when designing a program, typically making numerous errors. They test the correctness of the program by running it immediately and revising it until no errors arise (i.e., debugging). This approach is generally preferred over establishing correctness by proving it using a mathematical proof method or methods, such as proof by induction, prior to execution, as the latter may be time-consuming and ineffective.

Programmers may use a "reasoned" strategy (Merisio et al., 2021, p. 191), dedicating significant time to reasoning about the correct code before testing it. Such reasoning can serve as a foundation for proving a program's correctness, thereby increasing confidence and understanding of the program. Studies have shown that college students' programming skills are related to their logical reasoning skills (e.g., Graafsma et al., 2023). Other studies have also shown that engaging college students in programming can enhance their logical reasoning skills (e.g., Sayginer & Tüzün, 2023).

To enhance PSTs' (novice programmers') justification skills — essential for generating and justifying claims in PR-integrated geometry learning contexts — our study guided them to use reasoning while working on PR-integrated tasks, particularly in programming. They were encouraged to prove program correctness by drawing on their geometric and programming knowledge and skills. By examining the reasons PSTs provided to justify the correctness of programs, our study aimed to identify the specific geometric and programming knowledge and skills they employed during PR-integrated geometry learning activities.

Slavit et al. (2021) noted that students' STEM ways of thinking during interdisciplinary STEM learning activities can be observed in the claims and reasoning they present. Mathematicians have used proof assistants (e.g., *Coq*, Bertot & Castéran, 2013; *Isabelle/HOL*, Nipkow et al., 2002; and *Lean*, de Moura et al., 2015), which are programming language software, with various purposes such as constructing proofs, verifying the correctness of proofs, and generating new conjectures. To use these proof assistants, both mathematical and programming knowledge, as well as a solid understanding of logic, are required. The PR-integrated geometry learning activities we designed for the study were tested to see if they provided learners with opportunities to explore the connections between mathematics (geometry) and programming, which may have help them understand the use of programming in producing proofs.

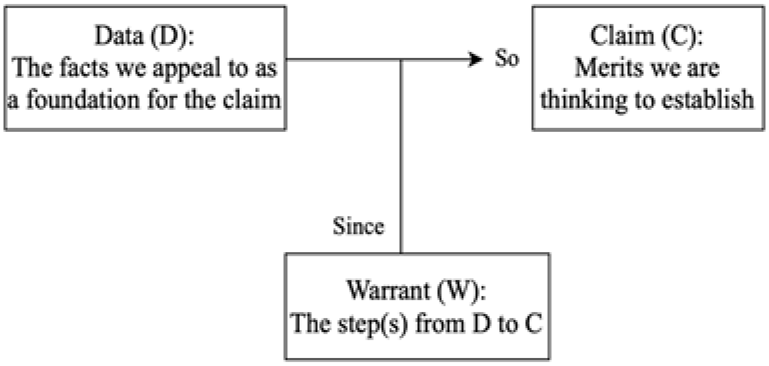
Description of the PR-Integrated Geometry Learning Activities

To support PSTs in learning to incorporate PR into mathematics classrooms, we designed PR-integrated geometry learning activities. These activities involve triangle exploration and construction tasks, especially developed for use in elementary and middle school classrooms. The tasks were developed based on research centered on effective mathematics teaching with PR (e.g., Humble et al., 2020; Vinnervik, 2022). The tasks for classroom use were peer-reviewed and published by NCTM (see the tasks included in Park et al.'s, 2023, article). For this study, we made slight modifications to the tasks to ensure they were appropriate for use in assisting PSTs' learning in a mathematics teacher education course.

To explore PSTs' learning through PR-integrated geometry learning activities, we implemented the activities across three sections of a geometry content course for K-8 PSTs over two semesters (two sections in the fall 2021 semester and one in the spring 2022 semester). The class intervention, which included the PR-integrated activities, occurred during the 2nd and 3rd weeks of each semester. The first author Park (a math teacher educator) served as the course instructor and recruited student participants from all three sections. At the university where our study was conducted, the geometry content course was the second in a sequence of content courses that K-8 PSTs were required to take. Topics in Numbers and Operations were covered in the first sequence course. The third sequence course covered fundamental concepts in algebra, probability, and statistics, such as sequences and functions, experimental and theoretical probabilities, and means and standard deviations.

Before engaging the PSTs in the PR-integrated geometry learning activities, during the 1st week of the semester, Park started lessons discussing with PSTs about defining a polygon, classifying different triangles, and constructing and evaluating mathematical arguments using Toulmin's argument model (Toulmin, 1958/2003), focusing on claims, warrants, and data (see Figure 1). They also discussed which arguments can count as proofs by comparing inductive and deductive reasoning used to support claims, with particular attention to the warrant components of arguments. We introduced Stylianides' (2007) conception of proof in school mathematics after the PSTs completed the PR-integrated geometry learning activities, to discuss further what arguments can be considered proofs in elementary classrooms.

Figure 1
Core Components of Toulmin's Argument Model (Adapted from Toulmin, 1958/2003)



During the 2nd week, after reviewing the topics discussed in the 1st week of lessons, we implemented PR-integrated geometry learning activities. On the 1st day of the class intervention, Park first had PSTs complete a presurvey that asked their thoughts on learning and teaching mathematics through PR. The second author, Boz (educational technology teacher educator), was invited to serve as a guest speaker to introduce PSTs to the use of block-based programming and educational robots at the elementary level. She joined the class virtually through the Zoom videoconferencing platform. Boz then gave a lesson on robot programming for 10-15 minutes to inform PSTs about how to program and operate a robot.

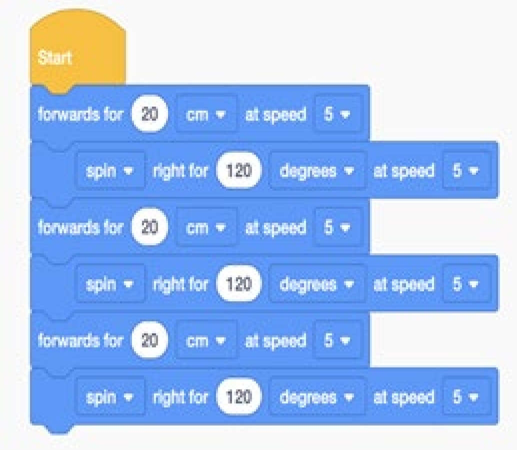
We used block-based programming (EdScratch, https://www.edscratchapp.com/) and physical robots (Edison Robots, https://meetedison.com/) to support PSTs' learning during their participation in PR-integrated geometry learning activities. Each PST brought their laptop for the activities so they could create and test robot programs themselves. One Edison robot was given to each PST during the activities. Before working on the triangle exploration and construction tasks, following guidance from Park and Boz, PSTs first had the opportunity to run their robots by programming them using *drive* blocks (e.g., *forward* and *spin* blocks) to explore how the robots would respond to the commands they received.

After this experiment, Park projected a program onto the screen so all PSTs in class could see it. The program was designed for Edison to trace the path of an equilateral triangle (see Figure 2), and PSTs were instructed to predict the program's output by carefully interpreting each line of code. PSTs first attempted to analyze the program individually, presenting arguments that described their interpretations and evaluations.

While constructing their arguments, PSTs were asked to consider using Toulmin's model of argumentation as a guiding tool, identifying data, warrant(s), and claim(s) in their arguments. They were then asked to share their arguments with their groupmates (three to four PSTs per group) and to post the coconstructed arguments they had collectively produced in a Google Slides electronic slideshow to display their work to the rest of the class. Using a ruler and protractor, each group then drew a shape on easel paper that is their anticipated program's output.

They programmed their Edison robots to test whether the robots could walk along the path of the shape they had drawn on paper. After these small-group activities, during the whole-class discussion activities, Park facilitated PSTs to share their groups' collective thoughts on (a) what lines of code are related to making three sides of a triangle, (b) why the robot needs to spin 120 degrees, and (c) what type of a triangle a robot would trace as the output of the program. During this discussion, Park and the PSTs further explored the geometric concepts of interior and exterior angles, as well as the angle measurements required for robot programming to ensure the robot turns by specific amounts.

Figure 2
EdScratch Program Designed for an Equilateral Triangle Construction



```
Start
forwards for 20 cm ▾ at speed 5 ▾
    spin ▾ right for 120 degrees ▾ at speed 5 ▾
forwards for 20 cm ▾ at speed 5 ▾
    spin ▾ right for 120 degrees ▾ at speed 5 ▾
forwards for 20 cm ▾ at speed 5 ▾
    spin ▾ right for 120 degrees ▾ at speed 5 ▾
```

On the 2nd day of the class intervention, Park had PSTs revisit the program for a robot to travel along the path of an equilateral triangle that they had previously explored and asked whether they could identify any patterns in the program. PSTs recognized a repeated pattern by pointing out lines of code that repeated. Park introduced PSTs to a *repeat* block within a *control* block in EdScratch, asking them to consider how they might revise the given program using the *repeat* block while still programming a robot to trace the equilateral triangle path.

Park encouraged PSTs to modify the original program using the *repeat* block in multiple ways. PSTs worked on this task with their group mates. Each group posted their revised programs with the *repeat* block through Google Slides to share with the class, and Park had PSTs evaluate the correctness of the programs from other groups. PSTs evaluated the programs developed by various groups by either operating their Edison robots or analyzing each line of code within the shared programs. After PSTs completed these activities, Park convened them to consider which of the two programs — those with and without the *repeat* block — would be more efficient.

After this discussion, PSTs moved to the subsequent small-group activities. First, in each group, PSTs were asked to draw an isosceles and acute triangle on easel paper using a ruler and a protractor and to measure each side and angle of the triangle. Drawing on their knowledge of programming and the triangle — including its side lengths and both interior and exterior angles — the PSTs were instructed to design EdScratch programs that would guide their robot to run along the path of the specified shape. All groups were also encouraged to create at least two programs using different blocks for their target shapes.

Before testing their programs with Edison robots, PSTs were asked to write arguments justifying their programs' correctness. That is, they were asked to elaborate on why they believed their programs would cause their robots to trace paths of target shapes before testing the programs' correctness with the robots. They were guided to write individual arguments on their worksheets and then engaged in small-group discussions to generate coconstructed arguments for their codesigned programs. Each group shared their work outputs via Google Slides to present to the rest of the class.

During the entire class discussion, Park encouraged PSTs to evaluate the correctness of the programs designed by different groups. They achieved this by either running a robot or analyzing the program line by line, then drawing an expected shape on paper based on their interpretation of the code. At the end of the lesson, PSTs were prompted to think further about whether it would be possible to construct all types of triangles, such as isosceles and obtuse triangles, using the *repeat* block, as they had for equilateral triangles.

At the conclusion of the 2nd day of the class intervention, Park assigned PSTs homework requiring them to design another EdScratch program for any quadrilateral of their choosing and to describe their program design process. As part of the homework, PSTs were also asked to reflect on their learning through the PR-integrated geometry learning activities. We asked this question again at the end of the semester, as part of the postsurvey, to examine their perspectives on using PR in mathematics classrooms after the class intervention.

Methods

To understand each participant's learning experience with PR during the PR-integrated geometry learning activities and the effectiveness of the PR intervention on their learning, we conducted an embedded single-case study (Yin, 2009) and used participants' artifacts collected during the activities as data for unique units of analysis. The selection of a single-case methodology is based on Yin's fourth rationale, with the study of the PSTs in the PR-integrated learning environment as a single *revelatory* case that is not typically available for formal investigation (p. 48). The inclusion of multiple units of analysis, organized at the participant level, suggests an embedded case study design (p. 50), rather than a holistic design intended to examine the global nature of the environment.

Since the data we collected for this study arose entirely from typical classroom activities, and we removed all identifying information for the participants, the study was deemed exempt from Institutional Review Board review under Title 45 Code of Federal Regulations 46.104, Categories 1 and 2. The data and results reported here arose in a "commonly accepted educational setting," under "normal educational practices," that were "not likely to adversely impact students' opportunity to learn."

Data Collection

For this study, we utilized the work completed by 15 volunteer PSTs [a] during their participation in PR-integrated geometry learning activities as part of their coursework. Our focus data included their written reflections and their in-class artifacts completed during the activities. We also used their responses from the pre-and postsurveys, which included open-ended questions, as supplementary data to understand what they brought into and out of the class.

Before the PR-integrated geometry learning activities began, we administered a presurvey to examine PSTs' initial perspectives about integrating PR into mathematics classes. This also helped us gauge their prior experiences and attitudes toward using PR in math education. Right after completing the activities, PSTs were assigned written reflections as homework. These reflections provided us with deeper insights into their learning through the activities they participated in. They offered valuable perspectives on ways the activities influenced their understanding of learning and teaching mathematics with PR.

At the end of the semester, a postsurvey was conducted to examine PSTs' attitudes toward PR integration and their willingness to incorporate PR in their future mathematics classrooms. During the activities, PSTs also produced various in-class artifacts that were tangible evidence of their engagement, learning processes, and application of geometric concepts in programming. Analyzing these materials helped us understand the learning opportunities that PR-integrated geometry learning activities offered to PSTs, enabling them to build the knowledge and skills necessary to incorporate PR in mathematics classrooms. Appendix A includes some example prompts used in the reflections and surveys. Although this variety of data sources helps triangulate our participants' experiences and improve the reliability of our data, the study's small sample size, the limited number of participants completing both the pre- and postintervention surveys, and the reliance on self-reported data limit the generalizability of our findings.

Data Analysis

To answer our research questions, we conducted an embedded single-case study (Yin, 2014). We explored the complexities and nuances of PSTs' experiences and learning processes by focusing on a single case. Our units of analysis were both at the participant and artifact levels. We examined individual initial and postsurvey responses, as well as written reflections, at the participant level. At the artifact level, we analyzed the outputs produced by the PSTs during the in-class activities. This includes all participants' individual arguments regarding the correctness of programs designed for robots to travel along triangle-shaped paths, as well as the arguments PSTs coconstructed with their group members during the small-group activities.

As part of our data analysis, each of the four authors carefully read the arguments PSTs had produced during the PR-integrated geometry learning activities. These arguments were part of their efforts to justify the correctness of programs before testing the programs with robots. Individually, we then analyzed the arguments using Toulmin's (1958/2003) model of argumentation as an analytical framework. The framework helped us identify the warrant components of the arguments, where we could see reasoning PSTs used in the program correctness verification processes, allowing us to identify the specific knowledge and skills they employed in these processes.

We recorded our findings in sets of analytic memos per participant. This claims-data-warrant (reasoning) process provided the guiding lens for each author's round of open coding (Corbin & Strauss, 1990). It served as a standard reference language used to bridge our areas of expertise (i.e., mathematics, educational technology, and computer science). In this process, each author compiled an evolving set of codes, filtered through the Toulmin model, and framed in terms of our individual areas of expertise. These codes were continually applied and expanded upon, taking into account each unit of analysis. The goal in this stage of our analysis was to develop a comprehensive yet nonexhaustive set of codes to describe the PSTs' engagement with PR-integrated geometry learning activities from various perspectives.

We then debriefed each participant's arguments, our analytic memos, and codes. We accepted commonly agreed-upon observations as significant and annotated these with unique codes to describe the standard features of the arguments. We employed axial coding(Corbin & Strauss, 1990) to identify significant events and landmarks across the units of analysis, thereby establishing a categorical organization and naming convention for the data. Once this list of codes was substantially complete, we returned to each set of participant arguments to verify that the coding scheme primarily described each participant's experiences.

The codes that emerged from the analysis of the arguments are presented in Appendix B, along with examples that describe each code. We used an open coding approach to examine PSTs' survey and written reflection responses, aiming to identify common themes in their evaluations of PR-integrated geometry learning activities and their learning experiences. We color-coded their responses as we read them line by line. We used an Excel spreadsheet as a tool to organize our color-coded data.

Results

In this section, based on our data analysis, we illustrate the specific knowledge and skills PSTs employed when asked to justify the programs' correctness, as well as the learning opportunities provided to PSTs during their participation in PT-integrated geometry learning activities.

Knowledge and Skills Used When Attempting to Justify Correctness of Programs

As described in the previous section, during the PR-integrated activities in a geometry content course, the first argument construction task asked PSTs to justify the correctness of a given program based on their own interpretations. The research team designed the program for a robot to trace an equilateral triangle path. The second argument construction task required PSTs to justify the correctness of a program they developed with their small-group members, which was designed to make a robot travel along a targeted acute-angled and isosceles triangle path. The final claims presented in their arguments were consistent across most participants, who argued that the programs would enable a robot to follow an equilateral triangle in the first task and an acute isosceles triangle in the second.

Based on our analysis of all of PSTs' arguments constructed for these two argument construction tasks, five PSTs made the final claims differently than the majority of PSTs, arguing that a program's output would make a "complete triangle" (Ivy) or "all types of triangles" (Mary). For the first argument construction task, Mary's group coconstructed their final claim, "The program can make all types of triangles," following Mary's. Similarly, for the second task, Ivy and Helen's group jointly made the final claim, "The robot's journey will make a complete triangle," following Ivy's initial claim. On the argument construction tasks, with some exceptions, most PSTs presented programs as the initial data that they would verify for correctness.

To connect their initial data and their claims to justify the correctness of the programs, most PSTs brought their geometric knowledge about definitions and properties of triangles and interior and exterior angles of 2D shapes they possessed, although some variations in the choices of what knowledge to use to support their claims in their arguments differed by PSTs and even differed by tasks. These patterns were observable in their coconstructed arguments produced with their group members. For instance, Helen provided the following warrants in her argument to justify the correctness of a program designed for a robot to follow an equilateral triangle path:

> We know that if a triangle has 3 equal sides, it is said to be equilateral. We also know that all triangles' angles have to add up to 180 degrees. So, in an equilateral triangle, because all angles are the same, and there are 3 of them, each angle should be 60 degrees. Since the angle measure is 120 degrees [in the given program], it is the measure of the exterior angle because 180 - 60 = 120.

However, we noted that in their written arguments on both the first and second argument construction tasks, a few PSTs did not clearly explain the specific geometric knowledge they used in programming (four PSTs in their first arguments and three in their second arguments). These PSTs instead made connections between program code and geometric constructs (e.g., sides and angles of a triangle) or at least described their understanding of each program code used in programming related to geometric constructs. See Kelly's interpretations of the program code as a representative example of how PSTs made these connections:

> The robot travels 5 cm to create the first side of the triangle. The robot then spins right for 100 degrees, as the exterior angle, to create the largest interior angle – 80 degrees. The robot then travels for another 5 cm and spins right 130 degrees, creating a smaller interior angle of 50 degrees. After traveling for 6.5 cm at the largest side of the triangle, then [it] spins right at 130 degrees, making an interior angle of 50 degrees.

PSTs' use of their skills in shape measurement and interpretation of the measurements was only observable in their arguments on the second argument construction task. Such a result makes sense because during activities on creating acute and isosceles triangles, participants were first asked to draw their target shapes using a ruler and a protractor. They were then tasked with designing programs based on the obtained measurements to make robots travel along the paths of their target shapes. In justifying their designed programs, seven PSTs used drawings of their target shapes, along with their measurements, as evidence.

The use of their skills in interpreting program code— focusing on repeated patterns — and in mental simulations of programs before testing their correctness using physical robots was evident in their arguments. However, there were some variations in their program interpretations or simulations. Interestingly, the use of mental simulations was less evident in their arguments on the second argument construction task. Based on their arguments in the first argument construction task, when justifying the correctness of the program designed for a robot to follow an equilateral triangle path, 10 PSTs (out of 15) used mental simulations to predict the program's output. However, in the process of constructing arguments for the second argument construction task, only three PSTs engaged in mental simulation of their designed programs to justify program correctness. No groups used mental simulations when coconstructing arguments during the second argument construction task.

Learning Opportunities Provided to PSTs During the PR-integrated Geometry Learning Activities

Our data analysis also identified three major themes that describe the learning opportunities offered by PR-integrated geometry learning activities. The following subsections illustrate each learning opportunity identified, using data as evidence, including PSTs' arguments, which they produced to justify the correctness of robot programs during the activities, and their feedback on PR-integrated activities submitted after completing the activities.

Learning Opportunity 1: Developing an Understanding of Geometric Concepts Used in Program Design

Thirteen PSTs (out of 15) evaluated the PR-integrated geometry learning activities as effective, highlighting an opportunity to deepen their understanding of geometric concepts, particularly the interior and exterior angles of 2D shapes. For instance, Mary responded, "I am personally a visual learner, and actually, being able to create a program that made my robot make a certain shape helped me to understand mathematical terms better, for example, exterior and interior angles" (written reflection). Similarly, Sarah stated,

> This exercise [PR-integrated geometry learning activities] definitely helped me learn and understand geometry better. When we had to put what degree the car (Edison) would turn each time, it really helped me understand the concept of exterior angles and how to find them. (written reflection)

Olia also mentioned, "They [PR-integrated geometry learning activities] really helped me understand triangles and interior angles."

The evidence of such learning opportunities was evident in the arguments PSTs constructed when asked to explain why their programs would enable robots to follow the paths of target shapes. They produced the arguments before testing the programs with robots. Appendix C includes samples of arguments that two PSTs generated during the activities. In their written arguments, these two PSTs specified which parts of their arguments comprised data, warrant, and claim components.

Olia's understanding of interior and exterior angles, as well as how to calculate exterior angles based on measurements of interior angles of her target shape, is explained in the warrant component of the argument she produced. Helen's application of her knowledge of angles in interpreting the given program, designed for a robot to travel along a path of an equilateral triangle, is evident in both the data and warrant components she provided in her argument. To verify the program's correctness, Helen checked if the three angle values entered were correct, calculating exterior angles based on her understanding of the relationship between an interior angle and its adjacent exterior angle. These examples illustrate how PSTs linked their geometric knowledge to program code to justify the program's correctness.

Learning Opportunity 2: Improving Justifying Skills Using Geometric Reasoning to Verify the Correctness of Robot Programs

All of the participants viewed *proving* as explaining why a statement is true. As a representative example, Aiden described its meaning this way: "Proving something is showing how or why something is the way it is or works in a more detailed and mathematical way" (Postsurvey). When asked about the features of a good mathematical argument acceptable as a proof, many emphasized that justification is its key component, making it crucial for an argument to explain why. For instance, Mary described in her postsurvey that "a good mathematical argument has an explanation and provides examples and evidence."

Eleven PSTs (out of 15) found the PR-integrated geometry learning activities helpful in improving their justification skills, particularly when they were asked to justify the correctness of their programs before testing them with their robots. While verifying the correctness of the programs, PSTs were guided to demonstrate why they believed their programs would enable robots to travel along the paths of the target shapes using geometric reasoning. So, following this guidance, they applied the geometric knowledge they had acquired to verify the correctness of the programs. For instance, Peter's group initially designed a block-based program that can make a robot trace an isosceles acute triangle path using the *repeat* block, but they later revised it without using the *repeat* block (see Figure 3 for the programs designed by Peter's group).

We thought that because the program had a repeat of the same distance and turn twice, it would make the triangle isosceles, and the follow-up code combined with the repeat code with degrees that kept the interior angles under 90 would make it an acute triangle. … [Later, after testing out their program using the robot] our group realized that the repeat pattern wouldn't work because it did not have the angles in the correct order.

In their original argument that Peter's group coconstructed, they provided the following as a warrant to support their claim about the correctness of their initial program with the *repeat* block:

> The rotation of 110 degrees and moving forward 5cm creates two sides of equal length and two equal angles. The last code created a different angle and side length. Altogether, the code met the requirements of creating an isosceles triangle with two equal sides and all angles under 90 degrees, creating acute angles.

In the case of Sarah's group, they expected that with their program (see Figure 4), the robot would travel an isosceles and acute triangle path that they created, providing the following justifications as warrants in their coconstructed argument:

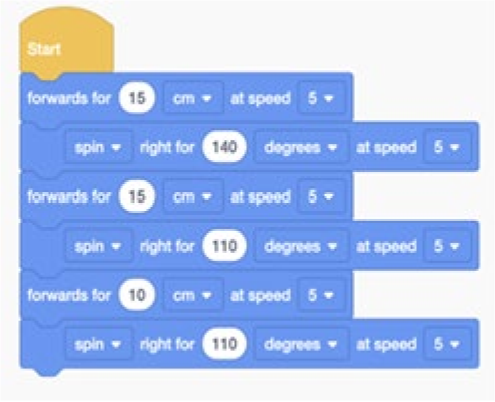
The robot moves forward 15 cm, making the first side, turns 140 degrees to make a 40-degree interior angle, goes forward 15 cm to make the equilateral side, turns 110 degrees to make a 70-degree interior angle, moves forward 10 cm to complete the base of the triangle, and then turns 110 degrees to have the robot facing the same direction it started. [An] isosceles and acute triangle is a triangle that has two sides of equal length and no angle greater than 90 degrees.

Figure 3
Two Programs Designed by Peter's Group With and Without the Repeat Block

```
Start
repeat 2
    forwards for 5 cm at speed 5
    spin right for 110 degrees at speed 5
forwards for 4 cm at speed 5
spin right for 140 degrees at speed 5
```

```
Start
forwards for 20 cm at speed 5
spin right for 140 degrees at speed 5
forwards for 20 cm at speed 5
spin right for 110 degrees at speed 5
forwards for 15 cm at speed 5
spin right for 110 degrees at speed 5
```

Figure 4
Program Designed by Sarah's Group



```
Start
forwards for 15 cm at speed 5
spin right for 140 degrees at speed 5
forwards for 15 cm at speed 5
spin right for 110 degrees at speed 5
forwards for 10 cm at speed 5
spin right for 110 degrees at speed 5
```

Peter's and Sarah's individually constructed arguments were not the same as their group's coconstructed argument, respectively, but their interpretations of each line of the code provided as warrants were connected to attributes of triangles they knew. Sarah noted the impact of PR-integrated activities on her learning in her written reflection as follows:

> It [PR-integrated geometry learning activities] also helped me make mathematical proofs because *when justifying why a triangle was a specific type of triangle or why a quadrilateral was, in fact, a quadrilateral, I could just look at the steps I had created for the car [robot] and use this information in my explanation*. (Italics added)

In her postcourse survey, Sarah mentioned, "Having to justify our programs was helpful for me. It helped me understand how to construct a proof." Similarly, Helen reflected, "I think that using the robots and the Ed Scratch programming makes it a lot easier to use proof and justification for things like making certain triangles or shapes, etc." Kelly said,

> I think that our class activities did help me engage with how to construct mathematical arguments. Before this class, I was a little bit clueless on what proofs would look like in a K-12 setting of math. I thought the Edison robots provided a clear example of why certain things happen the way that they do. It makes it easy to see what is happening and helps find patterns. (postsurvey)

Unlike other PSTs, Peter expressed the impact of the PR-integrated activities on improving his problem-solving skills "a little bit," adding his thoughts as follows: "The robots allowed me to provide a physical example of what I was arguing and provide another route of thinking for the problem" (written reflection).

Learning Opportunity 3: Building Productive Views on Learning and Teaching Mathematics with PR

Before participating in the PR-integrated geometry learning activities, nine PSTs admitted they had *no* prior programming experience. Four PSTs did not mention their prior experience with programming. Abielle and Peter shared that they had some programming experience but did not describe it in detail. Based on PSTs' reactions to the PR-integrated activities, it was evident that these activities were new to most of them. In their written reflections submitted after participating in the PR-integrated activities, 10 participants expressed positive attachments to their learning experience. Most PSTs said they "enjoyed" these activities and described the PR-integrated activities as bringing enjoyment and engagement in learning mathematics, as well as providing a new approach to learning. As an example, Helen wrote,

> *I really enjoy using the robots, and I think that it brings a new way of hands-on learning to the classroom.* If we, college students, are mesmerized by cute little robots, imagine how elementary- and middle-school-aged students would be when they see them. (Italics added; written reflection)

Ivy also stated, “I thought the robot activities were more engaging and hands-on for something that usually is not very involved and boring. This made me want to participate and get involved.” When asked to envision how learning and teaching mathematics through PR in their future classrooms would look, PSTs responding to this end-of-semester survey question also described similar scenarios, drawing on their positive PR-integrated learning experiences in class. They disclosed openness to integrating PR in their future elementary classrooms for the same reason.

However, two PSTs shared differing opinions, asserting that teaching programming may not be suitable for every student or that incorporating programming may not be relevant in every math lesson. For example, one of these PSTs addressed that although teaching PR “may be cool to teach it early, it may be more difficult as younger children may get more confused or distracted.” She viewed the inclusion of PR in math lessons as a “cool and interesting” approach, but noted that it can also be “time-consuming and confusing” for students.

Discussion

Our study showed that PR-integrated geometry learning activities created a meaningful learning space for PSTs. Specifically, the PR-integrated activities enhanced our PSTs’ understanding of geometric concepts, fostered positive attitudes toward teaching and learning mathematics with PR, and provided them with a PR-integrated learning context to build on their justifying skills and develop knowledge about the connections between programming and geometry.

The first learning opportunity focused on developing PSTs’ understanding of geometric concepts, particularly the angles of 2D shapes. By physically observing and controlling robots’ movements, PSTs transformed abstract geometric concepts into observable spatial actions. Most PSTs reported that PR-integrated activities helped them visualize and comprehend concepts such as interior and exterior angles and their relationship. This result aligns with prior studies that found that incorporating PR into mathematics lessons improved elementary students’ understanding of angles (e.g., Kim et al., 2021). One possible reason is that PR can help learners understand abstract mathematical concepts more concretely and accessibly (Bellas et al., 2019). Engaging PSTs in hands-on PR-integrated activities that require applying mathematical knowledge to program a robot can offer them opportunities to connect theoretical knowledge with practical implementation, thereby deepening their understanding of both mathematics and programming. This connection between tangible actions and conceptual understanding exemplifies the process of instrumental genesis (Dennis & Buchninder, 2023), in which tools like robots evolve from external artifacts into instruments that mediate mathematical thinking and reasoning.

The second learning opportunity focused on enhancing PSTs’ justification skills, which are essential in both programming and proving activities. When asked to construct arguments to justify the correctness of programs, most PSTs used their geometric knowledge, including definitions and properties of triangles and the interior and exterior angles of 2D shapes.

The use of Toulmin's (1958/2003) argument model helped guide PSTs in providing justifications, focusing on the warrant components of their arguments to support their claims. Thus, Toulmin's model can serve as an effective tool for PSTs to build justification skills, necessary for creating robust mathematical arguments and proofs. By pushing PSTs to justify the correctness of programs using reasoning, the PR-integrated activities also enabled them to deepen their knowledge of geometry and programming. Simply using programming in a mathematics classroom "does not in itself make the students see the connection to mathematical ideas" (Holo et al., 2023, p. 521). Such a learning experience can be compelling for PSTs' learning, as it can prepare PSTs to incorporate PR more efficiently and appropriately in mathematics classrooms drawing on the knowledge developed from the PR-integrated learning, particularly regarding the connections between programming and mathematics (geometry).

However, we also observed that although most PSTs explicitly made connections about how each line code is related to a specific geometric construct (e.g., a side or an angle of a triangle) when asked to justify the correctness of programs, in writing the arguments for the justifications, they did *not* focus much on elaborating their programming knowledge used in program design, such as why they arranged codes (blocks) in a particular order. Possibly, the program code they used was evident to them, so they did not feel the need to provide additional explanations. To make their learning more meaningful during their participation in our PR-integrated activities, PSTs need more practice presenting (explicit) warrants that draw on both their mathematics and programming knowledge, rather than focusing on using their math knowledge to verify the programs' correctness. By doing so, they could become more confident in programming, which in turn, could make teachers more comfortable incorporating PR into their mathematics teaching.

The third learning opportunity was developing productive views on learning and teaching mathematics with PR. The PR-integrated activities we designed to demonstrate a lesson model for teaching PR-integrated mathematics were effective. PSTs' views on learning and teaching mathematics with PR were overall positive, and most expressed openness to using PR in their future mathematics classrooms. This finding supports previous research suggesting that firsthand experience of the benefits of learning mathematics with PR may make PSTs more likely to adopt PR in their teaching of mathematics (e.g., Alqahtani et al., 2022). It also aligns with Silva et al. (2024), who emphasized that PSTs' perspectives and willingness to use PR evolve through guided practice and reflection.

The positive dispositions observed in our participants indicated not only affective engagement but also growing pedagogical awareness — a critical aspect of teacher education that prepares PSTs to integrate technology to support students' meaningful learning (Gadanidis et al., 2017). The enthusiasm expressed by PSTs also suggests that such activities can help them become more open to incorporating computer science into their future classrooms.

This work is especially relevant for those who work with prospective elementary teachers due to their general anxiety around mathematics (Gresham, 2007), their often naïve views of proof as procedurally generated structures that are void of explanatory power (Mingus & Grassl, 1999), and their largely female-identifying constituency, which has been historically marginalized in computing, coding, and information technology (Patitsas et al., 2014). Research in these areas, which our findings reinforce, indicates that formal experiences with these constructs, opportunities to interact with peers in their use and implementation, and the development of tools to support productive dispositions towards disciplinary practices alleviate PSTs' anxiety and encourage future engagement with topics that contribute to integrated learning and teaching. Our study provides a foundational approach to interdisciplinary learning and teaching, as well as to exploring 21st-century skills, which involve "strong communication and collaboration skills, expertise in technology, innovative and creative thinking skills, and an ability to solve problems" (Larson & Miller, 2011, p. 121).

Conclusion

Integrating PR into teacher education holds significant promise for enhancing PSTs' understanding of geometric concepts, improving their justifying skills, and fostering positive attitudes toward learning and teaching with PR. By providing hands-on, authentic learning experiences, PR-integrated geometry learning activities bridge the gap between theoretical knowledge and practical application. Teacher education programs should consider incorporating such activities to prepare PSTs for the demands of modern, technology-rich classrooms. This study also contributes to the growing body of literature advocating for the integration of computing into mathematics education. It highlights the importance of equipping future teachers with the knowledge, skills, and experiences to harness computer science (CS) to enhance mathematics learning outcomes (AMTE, 2022; Tamborg et al., 2022).

While the results are promising, several limitations must be acknowledged. The PR-integrated intervention was short, and it focused solely on PSTs' mathematics learning with PR in a geometry context. Exploring the impact of PR integration across other domains of mathematics can expand understanding of the potential benefits of its use on mathematics learning. Longer term studies are needed to assess the sustained impact of PR-integrated activities on PSTs' learning of mathematics and their future teaching practices. Future research should address these limitations by involving larger, more diverse groups of PSTs and by conducting longitudinal studies to examine long-term effects. Investigating how PSTs with PR experience design mathematics learning tasks that integrate PR, and observing their teaching during practicums or in their induction years of teaching — particularly how they teach mathematics and programming using these integrated tasks — could provide valuable insights regarding PSTs' preparations for integrated teaching.

Further research may explore how PR-integrated mathematics tasks function as formative assessment to gauge PSTs' mathematical and computational thinking in accessible, actionable ways. Studies might also identify which aspects of PR-integrated activities most effectively improve learning outcomes in both mathematics and programming. Understanding these elements can inform the design of PR-integrated teacher education and professional learning courses that support teachers' learning regarding effective integration of mathematics and CS in classrooms.

Notes

[a] All the data collected for this study was strictly followed by the procedures approved by the Institutional Review Board, and study participants' names used in this paper are pseudonyms. Mary, Ivy, and Helen were recruited from the morning section of the fall 2021 semester course, Emma, Olia, Kelly, and Sarah were from the afternoon section. In the spring 2022 semester, 8 PSTs (Amy, Daisy, Parker, Abielle, Lucy, Tucker, Peter, Aiden) were participated in the study. They were all from the same section of the class offered during that semester.

References

Abelson, H., & diSessa, A. (1981). *Turtle geometry: The computer as a medium for exploring mathematics*. MIT Press.

Ackerman, E. (2001). *Piaget's constructivism, Papert's constructionism: What is the difference?* Adapted from https://learning.media.mit.edu/content/publications/EA.Piaget%20_%20Papert.pdf?utm_source=chatgpt.com

Alqahtani, M. M., Hall, J. A., Leventhal, M., & Argila, A. N. (2022). Programming in mathematics classrooms: Changes in pre-service teachers' intentions to integrate robots in teaching. *Digital Experiences in Mathematics Education*, *8*(1), 70–98. https://doi.org/10.1007/s40751-021-00096-6

Association of Mathematics Teacher Educators. (2022). *Position of the association of mathematics teacher educators on technology*. https://amte.net/sites/amte.net/files/AMTE%20Technology%20Statement%20Oct%202022_0.pdf

Bellas, F., Salgado, M., Blanco, T.F., & Duro, R.J. (2019). Robotics in primary school: A realistic mathematics approach. In L. Daniela (Ed.), Smart learning with educational robotics (pp. 149-182). Springer, Cham. https://doi.org/10.1007/978-3-030-19913-5_6

Bertot, Y., & Castéran, P. (2013). *Interactive theorem proving and program development: Coq'Art: The calculus of inductive constructions*. Springer Science & Business Media. https://doi.org/10.1007/978-3-662-07964-5

Bieda, K., Ji, X., Drwencke, J., & Picard, A. (2013). Reasoning-and-proving opportunities in elementary mathematics textbooks. *International Journal of Educational Research*, *64*, 71–80. https://doi.org/10.1016/j.ijer.2013.06.005

Boz, T., & Allexsaht-Snider, M. (2022). How do elementary school teachers learn coding and robotics? A case study of mediations and conflicts. *Education and Information Technologies*, *27*(3), 3935–3963. https://doi.org/10.1007/s10639-021-10736-4

Bråting, K., & Kilhamn, C. (2022). The integration of programming in Swedish school mathematics: Investigating elementary mathematics textbooks. *Scandinavian Journal of Educational Research*, *66*(4), 594–609. https://doi.org/10.1080/00313831.2021.1897879

Clements, D. H., & Meredith, J. S. (1992). *Research on Logo: Effects and efficacy*. Logo Foundation. https://el.media.mit.edu/logo-foundation/resources/papers/pdf/research_logo.pdf

Corbin J., & Strauss A. (1990). *Basics of qualitative research: grounded theory procedures and techniques*. Sage Publications.

Dennis, C., & Buchbinder, O. (2023). Opportunities for instrumental genesis in mathematics lessons. In T. Lamberg & D. Moss (Eds.), *Proceedings of the 45th annual meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education* (Vol. 2, pp. 317–325). University of Nevada, Reno.

de Moura, L., Kong, S., Avigad, J., Van Doorn, F., & von Raumer, J. (2015). The Lean theorem prover (system description). In A. Felty & A. Middeldorp (Eds.), *Automated Deduction-CADE-25* (pp. 378–388). Springer International Publishing. https://doi.org/10.1007/978-3-319-21401-6_26

Department for Education. (2021). National curriculum in England: Mathematics programmes of study. https://www.gov.uk/government/publications/national-curriculum-in-england-mathematics-programmes-of-study/national-curriculum-in-england-mathematics-programmes-of-study

de Villiers, M. (1990). The role and function of proofs in mathematics. *Pythagoras*, *24*, 17–24.

Dogan, M.F., & Williams-Pierce, C. (2019). Supporting teacher proving practices with three phases of proof, *TEAN Journal*, *12*(1), 48–59.

Forsström, S. E., & Kaufmann, O. T. (2018). A literature review exploring the use of programming in mathematics education. *International Journal of Learning, Teaching and Educational Research*, *17*(12), 18–32. https://doi.org/10.26803/ijlter.17.12.2

Gadanidis, G., Cendros, R., Floyd, L., & Namukasa, I. (2017). Computational thinking in mathematics teacher education. *Contemporary Issues in Technology and Teacher Education, 17*(4). https://citejournal.org/volume-17/issue-4-17/mathematics/computational-thinking-in-mathematics-teacher-education

Goldenberg, E. P., Carter, C. J., Mark, J., Reed, K., Spencer, D., & Coleman, K. (2021). Programming as language and manipulative for second-grade mathematics. Digital *Experiences in Mathematics Education*, *7*(1), 48–65. https://doi.org/10.1007/s40751-020-00083-3

Graafsma, I. L., Robidoux, S., Nickels, L., Roberts, M., Polito, V., Zhu, J. D., & Marinus, E. (2023). The cognition of programming: Logical reasoning, algebra and vocabulary skills predict programming performance following an introductory computing course. *Journal of Cognitive Psychology, 35*(3), 364–381. https://doi.org/10.1080/20445911.2023.2166054

Gresham, G. (2007). A study of mathematics anxiety in pre-service teachers. *Early Childhood Education Journal, 35*(2), 181–188.

Grover, S., & Pea. R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Hanna, G. (1989). Proofs that prove and proofs that explain. In G. Vergnaud, J. Rogalski, & M. Artigue (Eds.), *Proceedings of the International Group for the Psychology of Mathematics Education* Vol. II (pp. 45–51). PME Books.

Hanna, G. (1990). Some pedagogical aspects of proof. *Interchange*, *21*(1), 6–13. https://doi.org/10.1007/BF01809605

Harper, F., Stumbo, Z., & Kim, N. (2021). When robots invade the neighborhood: Learning to teach preK-5 mathematics leveraging both technology and community knowledge. *Contemporary Issues in Technology and Teacher Education*, *21*(1), 19–52. https://citejournal.org/volume-21/issue-1-21/mathematics/when-robots-invade-the-neighborhood-learning-to-teach-prek-5-mathematics-leveraging-both-technology-and-community-knowledge

Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2018). A scoping review of studies on computational thinking in K–12 mathematics classrooms. *Digital Experiences in Mathematics Education*, *4*(1), 48–69. https://doi.org/10.1007/s40751-017-0038-8

Holo, O. E., Kveim, E. N., Lysne, M. S., Taraldsen, L. H., & Haara, F. O. (2023). A review of research on teaching of computer programming in primary school mathematics: Moving towards sustainable classroom action. *Education Inquiry*, 1–16. https://doi.org/10.1080/20004508.2022.2072575

Hoyles, C., Sutherland, R., & Evans, J. (1986). Using Logo in the mathematics classroom: What are the implications of pupil-devised goals? In P. R. Smith (Ed.), *Advances in computer assisted learning* (pp. 61–71). Pergamon. https://doi.org/10.1016/B978-0-08-031813-4.50015-4

Humble, N. (2023). A conceptual model of what programming affords secondary school courses in mathematics and technology. *Education and Information Technologies*. https://doi.org/10.1007/s10639-023-11577-z

Humble, N., Mozelius, P., & Sällvin, L. (2020). Remaking and reinforcing mathematics and technology with programming – Teacher perceptions of challenges, opportunities and tools in K-12 settings. *The International*

Journal of Information and Learning Technology, 37(5), 309–321. doi: 10.1108/IJILT-02-2020-0021

Israel, M., & Lash, T. (2020). From classroom lessons to exploratory learning progressions: Mathematics + computational thinking. *Interactive Learning Environments*, *28*(3), 362–382. https://doi.org/10.1080/10494820.2019.1674879

Kaufmann, O. T., Maugesten, M., & Meaney, T. (2023). The professional identities of prospective mathematics teachers who teach through programming. *Journal of Mathematics Teacher Education*. https://doi.org/10.1007/s10857-023-09594-3

Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education, 73*, 26–39. https://doi.org/10.1016/j.compedu.2013.12.010

Kim, C., Gleasman, C., Boz, T., Park, H., & Foutz, T. (2022). Learning to teach coding through argumentation. *Computers and Education Open*. https://doi.org/10.1016/j.caeo.2022.100107

Kim, Y. R., Park, M. S., & Tjoe, H. (2021). Discovering concepts of geometry through robotics coding activities. *International Journal of Education in Mathematics, Science, and Technology, 9*(3), 406–425. https://doi.org/10.46328/ijemst.1205

Knuth, E. J. (2002). Proof as a tool for learning mathematics. *Mathematics Teacher*, *95*(7), 486–490.

Lambic, D. (2011). Presenting practical application of mathematics by the use of programming software with easily available visual components. *Teaching Mathematics and Its Applications*, *30*(1), 10–18. https://doi.org/10.1093/teamat/hrq014

Larson, L. C., & Miller, T. N. (2011). 21st century skills: Prepare students for the future. *Kappa Delta Pi Record*, *47*(3), 121–123. https://doi.org/10.1080/00228958.2011.10516575

London, R. L. (1970). Proving programs correct: Some techniques and examples. *BIT*, *10*, 168–182. https://doi.org/10.1007/BF01936865

Lopez-Caudana, E., Ramirez-Montoya, M. S., Martínez-Pérez, S., & Rodríguez-Abitia, G. (2020). Using robotics to enhance active learning in mathematics: A multi-scenario study. *Mathematics*, *8*(12), 2163. https://doi.org/10.3390/math8122163

Manley, E. D., & Park, H. (2024). Developing secondary mathematics prospective teachers' confidence in using programming in mathematics classrooms: Benefits of the use of large language models. *AMTE Connections*, *34*(1). https://amte.net/sites/amte.net/files/ParkManley_Connections_Fall2024.pdf

Meschede, T., Haque, Z., Warfield, M. E., Melchior, A., Burack, C., & Hoover, M. (2022). Transforming STEM outcomes: Results from a seven-year follow-up study of an after-school robotics program's impacts on freshman students. *School Science and Mathematics*, *122*(7), 343–357. https://doi.org/10.1111/ssm.12552

Merisio, C., Bozzi, G., Datteri, E. (2021). There is no such thing as a "trial and error strategy." In M. Malvezzi, D. Alimisis, D., & M. Moro (Eds.), *Education in and with robotics to foster 21st-century skills. EDUROBOTICS 2021. Studies in Computational Intelligence* (Vol 982; pp. 190–201). Springer, Cham. https://doi.org/10.1007/978-3-030-77022-8_17

Mingus, T. T., & Grassl, R. M. (1999). Pre-service teacher beliefs about proofs. *School Science and Mathematics*, *99*(8), 438–444.

Mitchell, M. (2019). *Artificial intelligence: A guide for thinking humans*. Farrar, Straus and Giroux.

National Council of Teachers of Mathematics (2000). *Principles and standards for school mathematics*. Author.

National Governors Association Center for Best Practices & Council of Chief State School Officers. (2010). *Common core state standards for mathematics*. Authors.

National Research Council. (2001, January 23). *Adding it up: Helping children learn mathematics* (News release). https://www.nationalacademies.org/news/adding-it-up-helping-children-learn-mathematics

Nipkow, T., Wenzel, M., & Paulson, L. C. (2002). *Isabelle/HOL: A proof assistant for higher-order logic*. Springer Berlin Heidelberg.

Papadakis, S., Vaiopoulou, J., Sifaki, E., Stamovlasis, D., Kalogiannakis, M. & Vassilakis, K. (2021). Factors that hinder in-service teachers from incorporating educational robotics into their daily or future teaching practice. In B. Csapó & J. Uhomoibhi (Eds.), *Proceedings of the 13th International Conference on Computer Supported Education* (Vol. 2, pp. 55–63). Science and Technology Publications, Lda.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Park, H., Boz, T., Sawyer. A., & Willingham, J. C. (2023). Triangle explorations and constructions using robots. *Mathematics Teacher: Learning and Teaching PK-12*, *5*(116), 392–398.

Park, H., & Manley, E. D. (2024). Programming-integrated mathematics learning for future elementary teachers and non-STEM majors. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education* (pp. 1014–1020). ACM Special Interest Group on Computer Science Education. https://doi.org/10.1145/3626252.3630908

Patitsas, E., Craig, M., & Easterbrook, S. (2014, June). A historical examination of the social factors affecting female participation in computing. *Proceedings of the 2014 conference on Innovation & technology in computer science education* (pp. 111–116). https://doi.org/10.1145/2591708.2591731

Savard, A., & Highfield, K. (2015). Teachers' talk about robotics: Where is the mathematics? In M. Marshman, V. Geiger, & A. Bennison (Eds.), *Mathematics education in the margins. Proceedings of the 38th annual conference of the Mathematics Education Research Group of Australasia* (pp. 540–546). MERGA

Sawyer, A. G. (2024). Artificial intelligence chatbot as a mathematics curriculum developer: Discovering preservice teachers' overconfidence in ChatGPT. *International Journal on Responsibility*, *7*(1). https://doi.org/10.62365/2576-0955.1106

Sayginer, S., & Tüzün, H. (2023). The effects of block-based visual and text-based programming training on students' achievement, logical thinking skills, and motivation. *Journal of Computer Assisted Learning*, *39*, 644–658. doi: 10.1111/jcal.12771

Silva, R., Costa, C., & Martins, F. (2024). Pre-service teachers' perceptions towards integrating educational robotics in the primary school. *EURASIA Journal of Mathematics, Science and Technology Education, 20*(4). https://doi.org/10.29333/ejmste/14356

Slavit, D., Grace, E., & Lesseig, K. (2021). Student ways of thinking in STEM contexts: A focus on claim-making and reasoning. *School Science and Mathematics*, 121, 466–480. https://doi.org/10.1111/ssm.12501

Stylianides, A. J. (2007). Proof and proving in school mathematics. *Journal for Research in Mathematics Education*, *38*(3), 289–321.

Stylianides, G. J. (2008). An analytic framework of reasoning-and-proving. *For the Learning of Mathematics*, *28*(1), 9–16.

Suters, L., & Suters, H. (2020). Coding for the Core: Computational thinking and middle grades mathematics. *Contemporary Issues in Technology and Teacher Education*, *20*(3), 435–471. https://citejournal.org/volume-20/issue-3-20/mathematics/coding-for-the-core-computational-thinking-and-middle-grades-mathematics

Swedish National Agency for Education. (2020). *Curriculum for the compulsory school, preschool class and school-age educare* (Lgr22). https://www.skolverket.se/publikationer?id=13128

Tamborg, A. L., Elicer, R., & Spikol, D. (2022). Programming and computational thinking in mathematics education: An integration towards AI awareness. *KI - Künstliche Intelligenz*, *36*(1), 73–81. https://doi.org/10.1007/s13218-021-00753-3

Terwilliger, M. G., Jackson, J. L., Stenger, C. L., & Jerkins, J. A. (2019). Using computer programming activities and robots to teach generalization of a geometry concept. *Journal of Computing Sciences in Colleges*, *34*(3), 82–90. http://dl.acm.org/citation.cfm?id=3306465.3306478

Toulmin, S. E. (2003). *The uses of argument* (updated ed.). Cambridge University Press. Original work published in 1958.

Trouche, L. (2005). Instrumental Genesis, individual and social aspects. In D. Guin, K. Ruthven, & L. Trouche (Eds.), *The didactical challenge of symbolic calculators* (Mathematics Education Library series, Vol. 36). Springer. https://doi.org/10.1007/0-387-23435-7_9

Tsakeni, M. (2022). STEM education practical work in remote classrooms: Prospects and future directions in the post-pandemic era. *Journal of Culture and Values in Education*, *5*(1), 144–167.

Vegas, E., & Fowler, B. (2020). *What do we know about the expansion of K-12 computer science education? A review of the evidence.* The Brookings Institution.

Vinnervik, P. (2022). Implementing programming in school mathematics and technology: Teachers' intrinsic and extrinsic challenges. *International Journal of Technology and Design Education*, *32*(1), 213–242. https://doi.org/10.1007/s10798-020-09602-0

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

Yadav, A., Ocak, C., & Oliver, A. (2022). Computational thinking and metacognition. *TechTrends, 66*(3), 405–411. https://doi.org/10.1007/s11528-022-00695-z

Yin, R. (2009). *Case study research: Design and methods* (4th ed.). Sage Publications, Inc.

Zhang, C., & Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. *Journal of Industrial Information Integration, 23*. https://doi.org/10.1016/j.jii.2021.100224

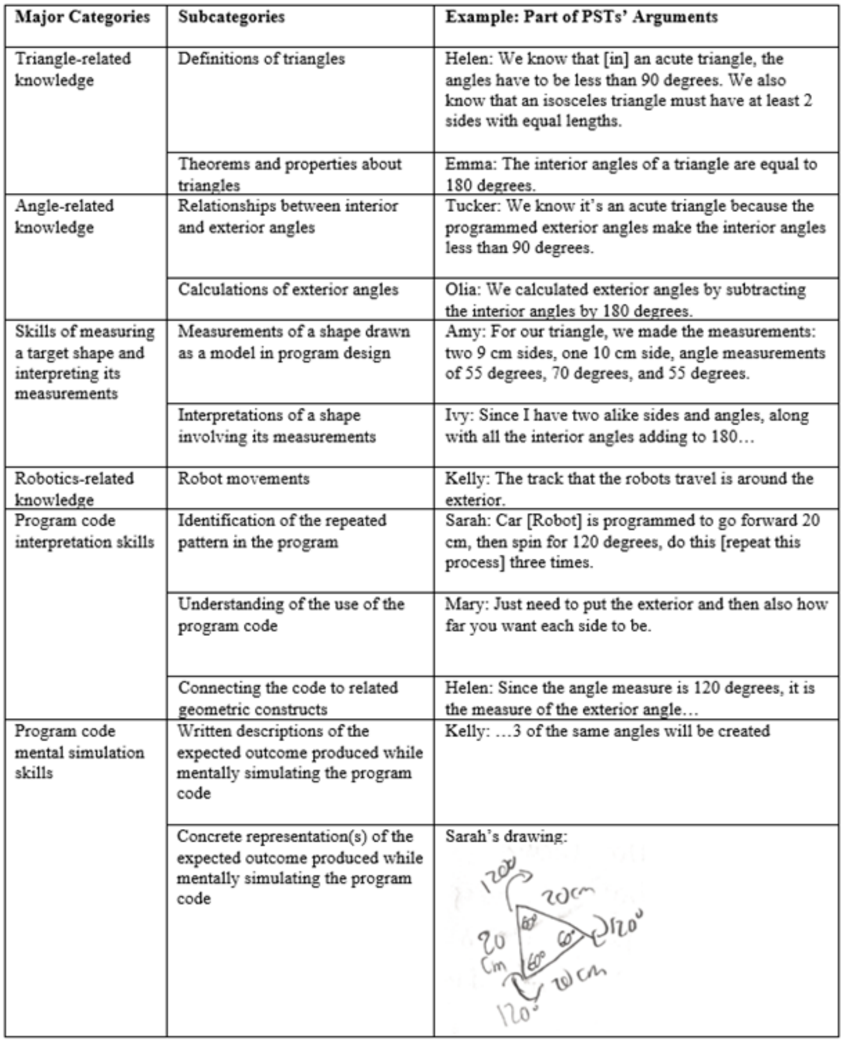
Zhong, B., & Xia, L. (2020). A systematic review on exploring the potential of educational robotics in mathematics education. *International Journal of Science and Mathematics Education, 18*(1), 79–101. https://doi.org/10.1007/s10763-018-09939-y

Contemporary Issues in Technology and Teacher Education is an online journal. All text, tables, and figures in the print version of this article are exact representations of the original. However, the original article may also include video and audio files, which can be accessed online at http://www.citejournal.org

Appendix A
Example Prompts Used in Written Reflections and Surveys

Instrument	Example Prompts
Presurvey (Completed *before* the PR-integrated activities)	- What does it mean to prove something? What would you say if asked to give a definition for proving? - Why do we prove things in math? How important do you think proving is in mathematics? - What do you think about the inclusion of programming and robotics (PR) in math classes? Have you ever had such an experience? If so, how was it?
Written reflection (Completed as homework *right after* the PR-integrated activities)	- How were your experiences learning mathematics through PR-integrated learning activities? Do you think the PR-integrated learning activities you participated in helped you understand geometric concepts and shapes? If so, how? If not, why not? - Do you think the PR-integrated learning activities enhanced your skill in making mathematical arguments (proofs)? If so, how? If not, why not?
Postsurvey (Completed *at the end* of the semester)	- Describe what specific aspects of learning to program Edison robots help you learn about constructing mathematical arguments (proofs). - What do you think about the inclusion of PR in math classes? - Do you think one can learn math better when it is integrated with PR? Why or why not? - What do you think about teaching math integrated with PR in elementary schools? - Would you be willing to include PR in your future math classes? Why or why not?

Appendix B
Codes Emerging From PSTs’ Arguments Focusing on Their Reasoning Used in the Process of Proving the Correctness of the Programs



Major Categories	**Subcategories**	**Example: Part of PSTs’ Arguments**
Triangle-related knowledge	Definitions of triangles	Helen: We know that [in] an acute triangle, the angles have to be less than 90 degrees. We also know that an isosceles triangle must have at least 2 sides with equal lengths.
	Theorems and properties about triangles	Emma: The interior angles of a triangle are equal to 180 degrees.
Angle-related knowledge	Relationships between interior and exterior angles	Tucker: We know it’s an acute triangle because the programmed exterior angles make the interior angles less than 90 degrees.
	Calculations of exterior angles	Olia: We calculated exterior angles by subtracting the interior angles by 180 degrees.
Skills of measuring a target shape and interpreting its measurements	Measurements of a shape drawn as a model in program design	Amy: For our triangle, we made the measurements: two 9 cm sides, one 10 cm side, angle measurements of 55 degrees, 70 degrees, and 55 degrees.
	Interpretations of a shape involving its measurements	Ivy: Since I have two alike sides and angles, along with all the interior angles adding to 180…
Robotics-related knowledge	Robot movements	Kelly: The track that the robots travel is around the exterior.
Program code interpretation skills	Identification of the repeated pattern in the program	Sarah: Car [Robot] is programmed to go forward 20 cm, then spin for 120 degrees, do this [repeat this process] three times.
	Understanding of the use of the program code	Mary: Just need to put the exterior and then also how far you want each side to be.
	Connecting the code to related geometric constructs	Helen: Since the angle measure is 120 degrees, it is the measure of the exterior angle…
Program code mental simulation skills	Written descriptions of the expected outcome produced while mentally simulating the program code	Kelly: …3 of the same angles will be created
	Concrete representation(s) of the expected outcome produced while mentally simulating the program code	Sarah’s drawing:

Appendix C
Sample Arguments from Two PSTs Constructed During the PR-Integrated In-Class Activities Considering Toulmin's Argument Model

Olia's Argument			
Activities	**Data**	**Warrant**	**Claim**
Verifying the correctness of the program PSTs designed for a robot to follow an acute and isosceles triangle path	1. Forwards 8.9 2. Spin right 140 3. Forward 8.9 4. Spin right 110 5. Forward 6 6. Spin right 110 	The sum of our interior angle[s] is 180 degrees. Because the robot draws on the outside, we calculated exterior angles by subtracting the interior angle by 180 degrees.	We know it will make a triangle when we apply our measurements to the program. It creates both [acute and isosceles] triangles.
Helen's Argument			
Activities	**Data**	**Warrant**	**Claim**
Verifying the correctness of the given program designed for a robot to travel an equilateral triangle path Given program: 	We know that if a triangle has 3 equal sides, it is said to be equilateral. We also know that all triangle angles add up to 180 degrees, so in an equilateral triangle, because all angles are the same and there are 3 of them, each angle should be 60 degrees.	Since the angle measure is 120 degrees, it is the measure of the exterior angle because 180 – 60 = 120.	So, because of this, the robot does, in fact, make an equilateral triangle.