2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #33383

Broadening the Middle School Computational Thinking Interventions Beyond Block Programming

Dr. Mohsen M. Dorodchi, University of North Carolina at Charlotte

Dr. Dorodchi has been teaching in the field of computing for over 30 years of which 20 years as an educator. He has taught the majority of the courses in the computer science and engineering curriculum over the past 20 years such as introductory programming, data structures, databases, software engineering, system programming, etc. He has been involved in a number of National Science Foundation supported grant projects including Scholarship for STEM students (S-STEM), Researcher Practitioner Partnership (RPP), IUSE, and EAGER.

Alexandria Benedict, University of North Carolina at Charlotte

Alexandria Benedict is a graduate student at the University of North Carolina at Charlotte pursuing her Master's in Computer Science. She is a research assistant under the RPP STEM Ecosystem Project which helps study the effects of computational thinking inside classrooms.

Audrey Rorrer
Dr. David K. Pugalee, University of North Carolina at Charlotte

Dr. David Pugalee is a full professor, and Director of the Center for Science, Technology, Engineering, and Mathematics Education (STEM) at UNC Charlotte. The recipient of millions of dollars in grant-funding, Dr. Pugalee has also published works on STEM teaching and learning including recent books Lesson Imaging in Math and Science and Effective Content Reading Strategies to Support Scientific and Mathematical Literacy. Dr. Pugalee has also worked with multiple STEM education projects including the current IES project 5E Model Professional Development in Science Education for Special Educators and the NSF Project, Developing a Systemic, Scalable Model to Broaden Participation in Middle School Computer Science which focuses on computational thinking in science and mathematics.

Dr. Lijuan Cao, University of North Carolina Charlotte

Broadening the Middle School Computational Thinking Interventions beyond Block Programming

Mohsen Dorodchi, Alexandria Benedict, Audrey Rorrer,
David K Pugalee, Lijuan Cao, Mary Lou Maher

Abstract

The contributions of this research paper are as follows: 1) it builds on prior work done in transitioning teaching and learning of programming and computational thinking from block-based programming to text-based programming languages, and 2) it infuses this model of programming to teach computational thinking (CT) in middle school curriculum and studies the impact of the professional development (PD) on teachers. Our proposed curricular approach is based on the request from our partner middle school teachers in exposing students to Python. Through the existing researcher-practitioner partnerships (RPPs), we have been providing CT professional development for the teachers based on block programming. Furthermore, we developed and presented the new materials in two online PDs as a short and a more detailed one in Spring and Summer of 2020 to help the teachers with planning for the 20-21 school year. Moreover, we studied how such PD would impact the teachers.

Our prior study of the middle school ecosystem revealed that teachers' needs should always be the focal point of PDs, as there are many unmet needs for CS/CT in K-12. Our first challenge was to provide a holistic view of programming in Python while reassuring the teachers that they are able to learn the language and teach it to their students. Furthermore, integrating CT into school curriculum poses new challenges regarding the changes. We find that the available and existing resources for transitioning from block programming to Python require further insightful reconfiguration and modification to fit the middle school curriculum.

1 Introduction

Computational thinking (CT), as a set of fundamental skills that support problem solving and understanding human behavior [1–3], continues to receive consideration as a critical component of middle school programs. This emphasis is important given that computational thinking has primarily focused on computer science with little emphasis in other disciplines [4, 5]. Challenges in integrating computational thinking into the school curriculum necessitates changes in policy including addressing significant issues around infrastructure, and providing teachers the resources that develop a cogent understanding of computational thinking as well as relevant and appropriate exemplars of age appropriate cases [6]. Such focus would promote core concepts essential to effective computational thinking development such as designing solutions to problems through

abstraction, automation, algorithmic thinking, data collection and data analysis; implementing designs; testing and debugging; modeling, running simulations, conducting systems analysis; reflecting on processes and communicating ideas; recognizing abstraction and moving between levels; innovation, exploration and creativity; problem solving and use of differing and multiple learning strategies. Research has demonstrated that simple design projects positively impact students' computational thinking including problem representation, generation and implementation of solutions, exploration of multiple potential solutions, and problem-solving on multiple levels [7, 8].

To enhance the CT in K-12, professional development (PD) is critical to support teaching and to enhance student learning. Authors in [9] reviewed thirty-five studies on professional development to identify successful components that are tied to student learning gains. These components included content focused on incorporate active learning, engage teachers in collaboration, use of models or modeling, expert support, feedback and reflection, and sustained duration. A collaborative culture that builds professional capital of teachers promotes widespread improvement in an organization [10]. These perspectives support what teachers identify as important to their professional development: interaction, engagement relevant for their students; provide practical ways to deliver instruction; be teacher-driven; and sustainable [11]. Professional development for middle and high school teachers that include CT principles through a 3C model of *code, connect,* and *create* has demonstrated shifts in teacher understanding of the role of computational thinking as it relates to their content as well as their own self-efficacy related to CT integration into their disciplines [12].

Research on technology-related professional development with middle school science teachers showed growth in teachers' technological literacy, their capabilities with technology, and pedagogical beliefs which was correlated to students' performance [13]. Other research shows that relatively short but targeted professional learning can improve teachers' computational thinking, pedagogical capabilities, technological know-how, and confidence [14]. In general, effective professional development for teachers on computational thinking in relation to programming is effective when there is a focus on fundamental subject knowledge of programming for development computational thinking and when there are opportunities for practice and reflection on practice [15]. Their research with seventy-six teachers reported that the teachers developed deeper understanding of computational thinking concepts and practices and improved in their content knowledge of programming for computational thinking development, technological content knowledge on use of block-based programming environments, and the application to use programming for computational thinking with appropriate pedagogical tools in subject contexts.

In this paper, we first review the literature related to the following three distinct areas of this study: 1) online PD, 2) Middle school CS and CT, and 3) transitioning experiences from block-based to text-based programming languages. Next, we present our study followed by the results and discussions about them with the future direction.

1.1 Middle school CS and CT

There are many unmet needs in K-12 Computer Science education despite it's high demand. According to [16], only a quarter of schools have a dedicated CS teacher, with even less exposure to minority demographics. Providing these teachers with adequate resources and support is one way to address this concern and lessen the gap in CS education. Many teachers and parents are supportive of exposing students early to CS concepts, as computational thinking can be integrated into many different subjects and provide students with broader problem solving skills through algorithmic thinking. A solution to this problem is to provide K-12 teachers with many PD opportunities of varying formats.

One method of PD for expanding K-12 CS education was deployed and studied in the Science and Technology Excellence Program (STEP) in Israel [17]. As a program originally implemented to increase the number of students pursuing STEM fields in higher education, one of STEP's main focuses is on middle school CS curriculum for students, specifically computational thinking concepts and algorithms. This CS program was developed to span three years of MS education, with each year focusing on a specific goal. These goals mentioned, in order, is as follows: 1) core concepts of computational thinking and programming in Scratch, a block-based programming language known for its interactivity and ease of use, 2) methods and tools of scientific research in STEM, 3) an elective area of study focusing on previous concepts taught, and 4) a final programming project of the students' choice. As emphasized by authors of this research, with this program came the difficulty of finding qualified teachers who could teach CS at the middle school level [17]. In order to address this issue, STEP implements an elaborate training program to provide new teachers with pedagogical tools for their classrooms. Feedback was later gathered from these teachers on its effectiveness. The training program consists of several courses, each being approximately 3 months long, to teach educators the concepts needed for the modules of the middle school CS program. Furthermore, teachers were provided with both online and in-person instruction. A total of sixty teachers completed the program and were surveyed on the outcome of training and satisfaction of their students' outcomes. The study concludes a resulting high satisfaction among teachers and positive feedback regarding the amount of support provided by the facilitators, specifically in regards to an online forum given as a virtual support tool during and after the time of training. By providing teachers with extensive training, various forms of instructional methods, and resources to give teachers continuous support even after training completion, the authors successfully implemented a middle school CS curriculum with high teacher and student satisfaction.

1.2 Online PD

Online professional development (PD) opportunities for K-12 CS teachers is an area that can be very beneficial in terms of addressing teacher needs. Online opportunities for teachers tend to be more cost-effective by preventing the need for travel, allowing easier access and thus being able to account for a larger population of CS teachers. Furthermore, if implemented successfully, the effectiveness of online PDs can match those of formerly in-person PDs and allow for more support to teachers even after workshops are conducted, creating a community of teachers with similar needs. In the following related works, methods of instruction are provided along with the resulting teacher feedback.

In [18], a study was conducted to find a solution for formerly in-person K-12 PD workshops in response to the COVID-19 pandemic lockdowns. Researchers tested an online collaborative classroom tool to conduct their online PD for a curriculum aiming to prepare students for the secondary school Computer Science (CS) final examination held in Ireland. The online PD session for teachers was divided into three parts: 1) an online presentation of the concepts being taught, 2) breakout sessions for teachers to collaborate together and solve the algorithm, and lastly, 3) a discussion of the solutions found. The teachers who attended this online PD were surveyed afterward to provide feedback on the effectiveness of this delivery method. The study found that a majority of teachers responded positively to the PD and felt that the online delivery was just as effective as the previous in-person PDs they have attended [18]. Furthermore, by providing teachers with this method of online delivery for the PD, a similar format of online teaching could be reflected in teachers' classrooms in the future, if remote learning for students persists. This study conducted pertains closely to our own online PD for middle school teachers and displays a similarity in teacher sentiment to the challenge of short-notice transitioning to an online learning format.

Another work by [19] conducts an in-depth comparison of the effectiveness of an online CS PD versus a hybrid PD. Both PDs span a 4-week period and differ in two ways: 1) the online PD would be held virtually with video conferencing and email as communication, and 2) the hybrid PD would meet for half of the allotted time in-person. The online PD session consisted of approximately 10 teachers and one lead teacher, one who has experience teaching the designated CS course for at least one year. When surveyed afterward about their experiences with the PDs, about half of the attendees stated that they preferred the online format over the hybrid, and 29% having no preference between online or hybrid [19]. In addition to these results, a final exam was given to teachers to test the concepts taught within the PD, and both groups performed very similarly. These results show that an online delivery of PD content to teachers can be just as effective as in-person instruction despite the obstacles which can come with online collaboration.

1.3 Transitioning from block-based to text-based programming languages

In [20], the high school students' view of blocks-based programming was studied. In particular, the study was focusing on 1) do students perceive block-based easier and what are the features they identify as contributing to the perceived ease-of-use of block-based programming tools, 2) what are their perceptions on "the most salient differences between block-based and text-based programming", and 3) what are the drawbacks of block-based programming from their perspective? To answer the first question, students worked for five weeks in Snap! and then another five weeks learning Java followed by a survey. Over half of students found block-based programming easier. Furthermore, through interviewing 9 students from grades 8-12, the following items were identified as the reason for "ease-of-use" of blocks in programming: a) they are easier to read, b) the shape and graphical cues help with how and where they can be used, c) they found it easier to compose and create programs with blocks, and d) blocks do not need much memorization as it is required for the text-based programming syntax. In addition to these four items, the authors found additional differences frequently repeated in students surveys (to answer research question 2 above) as: e) how Java was not as conducive to the use of trial-and-error

programming, f) lack of prefabricated commands in text-based programming, and g) there were more items that were discussed in papers but not as frequent. Finally, related to the third research question, students mentioned the following items as the major drawbacks of block-based programming as: 1) Less Powerful (less set of things are possible with the block-based tool), 2) Slower Authoring and More Verbose (time and number of blocks it takes to compose a program in the blocks-based interface compared to the text-based alternative), and 3) Inauthentic (how closely the block-based programming tool and practices adhere to conventional, noneducational programming contexts). The authors conclude that with these findings we can make teachers aware of these items to be able to address them in their teachings as well as tool designers can provide new features to address students' concerns. Moreover, we found these findings an opportunity to improve our PD for the teachers.

In another work, authors conducted research on students' misconceptions about loops as one of the fundamental programming concepts among 207 elementary school students. These students were learning to program in three programming languages: Scratch, Logo and Python [21]. Authors introduced abstraction from literature as a core programming concept as well as major obstacle for novice programming learners [22], [23]. Furthermore, they discussed about cognitive development of children based on Piaget theory of cognitive development [24] which states that K-12 students are mostly at concrete or pre-formal phase of cognitive development and not in the abstract thinking phase. The authors then present a quantitative research with pre-test and post-test. The pre-test is used to determine students' problem-solving abilities and post-test to assess students' understanding of the sequencing and loop concepts with the focus on the loop concept. One other major contribution of their work is the theoretical discussion about the differences in learners as it relates to visual vs. text-based programming. In short, the concrete learners (aka "digital natives" [25]) like the trial and error feature of visual programming [26, 27] whereas abstract thinkers like the hierarchy and abstraction while programming which makes the text-based languages their preferred language.

2 Study Design: PD for middle school teachers

The growth trend of infusing computational thinking into K-12 curriculum is now impacting the students and teachers. There are more students who come to middle school familiar with block programming. Because of this pattern along with the growth of Python programming as an introductory programming language, the middle school teachers in a local CS Magnet school desired to learn Python so that they could better prepare their students for high school CS courses. We were asked to prepare and provide professional development workshops to teach teachers Python. In particular, we conducted two PDs over the Spring and the Summer of 2020. While the first PD covered an overview of the Python language, the second one was digging deeper into Python programming.

Our research questions guiding the study of our PD are

1. What benefits do teachers perceive in learning Python for themselves and for their students?
2. What challenges do teachers see for applying Python in their classes?
3. How effective was the PD workshop overall for the teachers?

To address these research questions, we provide context about the structure of the PDs, describe our approach to developing the curriculum for the PD workshop, and provide results from teacher surveys and undergraduate student assistants helping us facilitate the PD.

2.1 Methodology

To address the research questions, a short reflection survey was captured using Padlet, an online feedback tool, during both PD sessions. Teachers were informed of the research activities at the session welcome and given informed consent documents. Participation in the online surveys were voluntary. Additionally, a survey of self-efficacy in applying Python in their classes was given at the end of the Summer PD. 11 teachers participated in the online Padlet surveys, and 7 participated in the post-survey in Summer. Thematic analysis was conducted on the Padlet surveys, deploying an open coding process [28]. Descriptive statistics were performed on the post-survey to show means across items. Survey items were rated on a Likert-type scale, with the maximum score of 5 being "Strongly agree" and the minimum score of 1 being "Strongly disagree" to the provided statements about the workshop and confidence.

2.2 Spring PD

The first PD took place over the span of a half day at the end of the Spring semester. Due to Covid-19, the workshop was conducted online, where teachers connected through a group video conferencing using Zoom. Over 60 teachers and school academic staff participated in the workshop.

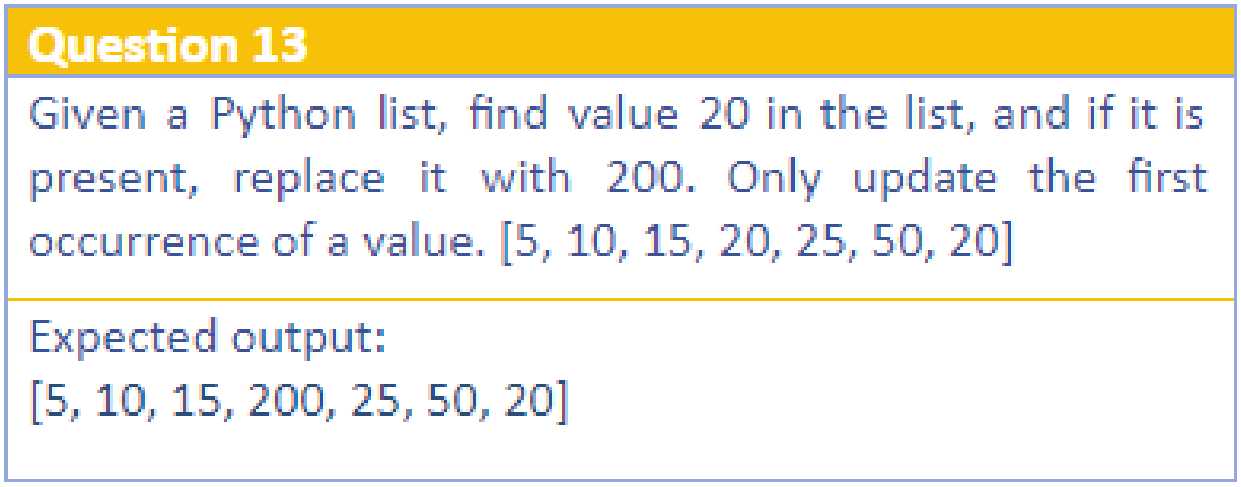
2.2.1 Curriculum

Since this is the first time for most of the participants to learn Python, the curriculum primarily focused on fundamental concepts, including variables, data types, decision making, loops, functions, and list/arrays. The workshop included 3 sessions. The first session introduced general knowledge about Python and why it is growing to be the first choice in programming. At the end of the first session, the participants were divided into small groups, each of which contains 6 participants and 1 teaching assistant. For the second session, each group joined a breakout room, where they worked on a number of hands-on programming exercises, ranging from simple to hard. For each exercise, there are step-by-step instructions along with screenshots of codes. An example of a programming exercise is given in Figure 1. The participants were encouraged to try the exercises at their own pace, to ask questions and to discuss with each other. Note that the participants completed the exercises using the online Python IDE Repl[1], which is a text-based programming environment.

During the third session, we had a closing discussion with the participants. Further, we brainstormed how we can better prepare for future workshops.

[1] https://repl.it/

Figure 1: Short programming exercise provided during the Spring PD



2.2.2 Discussion

Based on teachers' feedback, the participants overall found the PD beneficial; however, many noted the need for more time in mastering the fundamentals of text-based programming. They felt that, particularly for beginners, learning and understanding the concepts provided in the PD required more future workshops to build off of what they learned. Several teachers also mentioned that compared to their previous experience in block-based programming, remembering the functionality as well as syntax of specific text-based code was difficult and would require more practice for them to gain confidence with programming.

2.3 Summer PD

The second PD conducted over summer was more extensive, spanning over two days, and focused on a small group of lead teachers. It was also hosted online using Zoom. Eleven teachers attended the workshop divided into 4 breakout rooms with 4 TAs moderating the hands-on sessions. The first morning was focused primarily on block-based coding whereas the first afternoon and the next morning focused on introductory and more advanced concepts of Python followed by open lab session on the afternoon of the second day focusing on possible lesson plans for different classes.

2.3.1 Curriculum

When designing the Summer PD, we took the feedback from the Spring PD into consideration. Since most of the teachers had previous experience with a block-coding environment, e.g., Scratch [2], we wanted to help teachers learn Python by providing a transition from block programming to syntax-based coding. To facilitate this, we designed the Summer PD with two phases, phase 1 introduces concepts of programming using a blocked-based coding environment, and phase 2

[2] https://scratch.mit.edu/

focuses on text-based coding environment. In particular, for phase 1, we used EduBlocks [3], which is an open source web-based programming environment. It provides a block-based programming editor that allows one to program Python code on the blocks. Further, with one button click, it can switch the block code into Python code. Figure 2 shows an example of block-based code and Figure 3 shows the corresponding Python code and the editor. The examples demonstrate the interface of EduBlocks as well as the option to toggle between block-based programming and text-based programming. This features makes it easier for someone with only blocked-based coding environment experience to learn Python without worrying about the syntax. For phase 2, we used Trinket [4], which is an open source web-based Python compiler.

Figure 2: An example of block-based code within EduBlocks

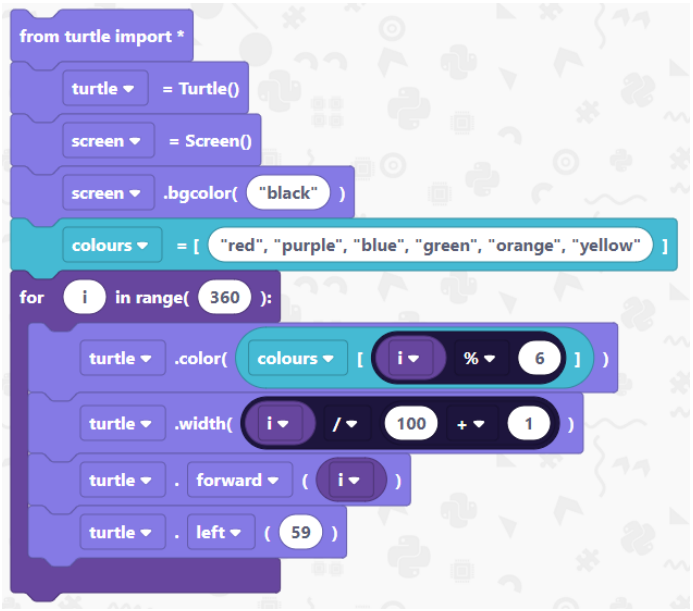


Figure 3: Corresponding Python Code of Figure 1a.

```
1  from turtle import *
2  turtle = Turtle()
3  screen = Screen()
4  screen.bgcolor("black")
5  colours = ["red", "purple", "blue", "green", "orange", "yellow"]
6  for i in range(360):
7      turtle.color(colours[i % 6])
8      turtle.width(i / 100 + 1)
9      turtle.forward(i)
10     turtle.left(59)
```

The first day was dedicated to introducing teachers to EduBlocks and teaching Python syntax

[3] https://app.edublocks.org/#Python
[4] https://trinket.io/features/python3

Table 1: PD Topics

Topic No.	Topics
1	Introduction: Getting Started
2	The EduBlocks Environment
3	Basic Concepts of Python
4	Simple Programs
5	Conditional statements / Iteration
6	Arrays and Lists
7	Advanced Concepts

through this tool. The basics of Python were taught in block-based programming including more advanced concepts such as iteration and loops, conditional statements, and definitions. By beginning with block-based programming, teachers are able to focus more on the actual concepts being taught rather than needing to also memorize complicated syntax and debug possible compilation errors. On the second day of the online PD, teachers were then presented with the same concepts in order of the first day, except they used text-based programming. Since they had already been exposed to the Python concepts the day before, teachers could then focus more on how to transition this knowledge to text-based programming. An example output for one of the programming activities teachers completed in the PD is shown in Figure 4 which uses the imported pygal library for pie chart creation. It's corresponding block code is shown in Figure 5. The PD flow of the topics are shown in Table 1. We discuss their actual experience later in the section focusing on the feedback.

Figure 4: Example output of an activity teachers completed in the PD

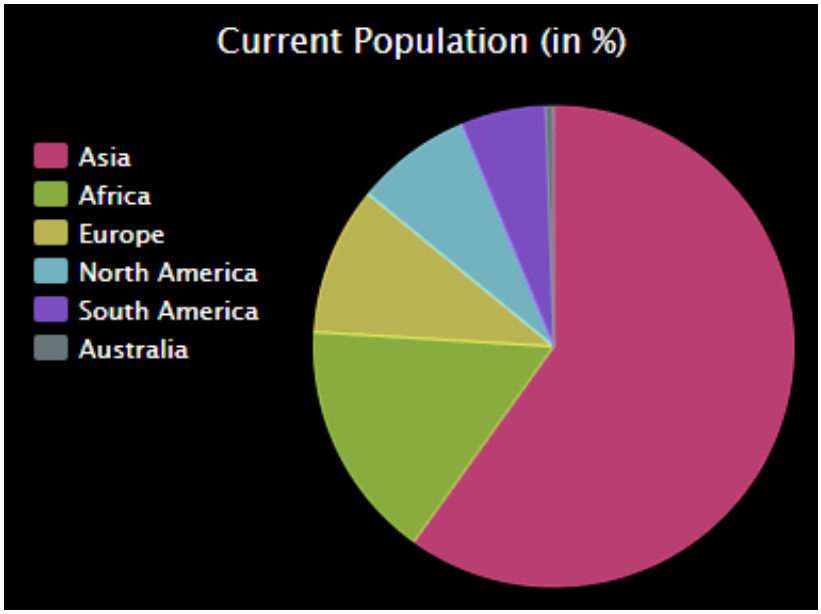
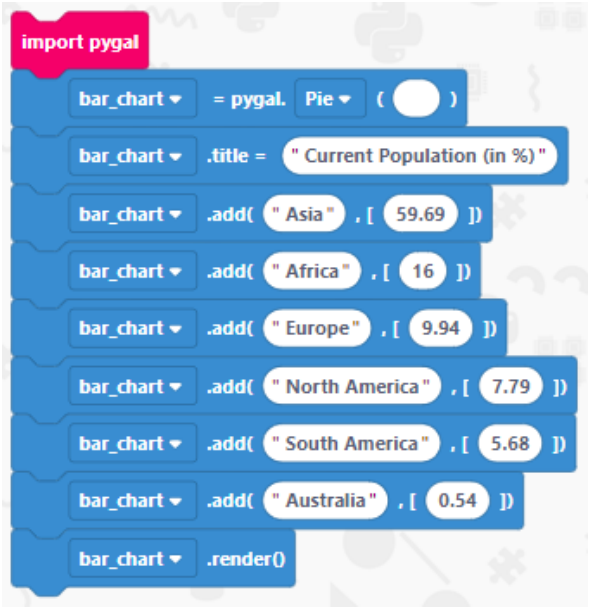


Figure 5: Corresponding block code to the pie chart activity



2.3.2 Discussion

Teachers who attended the Summer PD showed an overall high satisfaction with the PD when answering the questionnaire provided at the end of the workshop. Attendees enjoyed the activities given, felt engaged throughout the PD, and felt more confident in their abilities to learn Python. Similarly to the Spring PD, however, teachers still displayed uncertainty about their ability to apply the concepts learned in their own classrooms, and that more time would be needed to do so.

Python is traditionally taught in introductory CS courses, over the course of an entire semester. Condensing the training into sequential workshops, to provide instruction to teachers without any prior exposure, was a directive from the teachers themselves. It was unclear as to how effective online workshops would be in teaching Python to the teachers, and fostering confidence in the teachers in their ability to apply what they learned in their classes.

3 Analysis of the Results

In this section, we present the feedback we received from the participants from the middle school as well as the TAs helping out with the small group activities. In general, the teachers found the workshops very useful and they liked the overall convenience of the online offering. However, they faced several different challenges due to the online format of PD. Since the online format was new for the teachers in attendance, there was a slight learning curve in getting accustomed to the technology used, such as the video conferencing application used to conduct the PD. Furthermore, a few teachers noted connection troubles during the workshop, causing an inability to view the TAs screen sharing.

The thematic analysis performed on feedback revealed two emergent themes about time and

benefit. Teachers felt the need for more time to learn the materials with a 31.25% frequency. They also report that the PD was beneficial, with the frequency of 37.5% as shown in Table 2 for Spring PD.

The feedback from teachers expressing a preference for more time during the workshops is prevalent through both Spring and Summer PDs, as they spanned one to two days time which many would not consider enough instruction for learning the Python programming language. The PDs, however, had introduced teachers to the basics of Python with very good satisfaction ratings. A total of 7 teachers consented to taking the survey, with the mean, median, and standard deviations of their responses shown in Table 3.

Teachers had very positive ratings for most of these statements. Also noting that the survey was in response to the online format of PDs, teachers still felt that they had a great experience attending the workshop. When being asked whether they felt confident in their ability to eventually teach Python, teachers, on average, agreed to this statement with a positive rating (4). Furthermore, teachers felt that they could network with their colleagues despite being within an online format, showing that these methods did not detract from the sense of community that is typically provided by in-person PDs. The statements which teachers scored lower on the likert-scale included: having enough time within the workshop, feeling confident in using and teaching Python, feeling confident writing code, and incorporating and instructing students on applying Python concepts. These questions that scored lower show a common theme of feeling confident using and teaching the concepts in the present-tense, meaning that two workshops of Python instruction is insufficient to provide teachers with the support needed in preparation for classroom applications. Although teachers scored lower on feeling confident teaching the concepts learned, they alternatively scored higher on future teaching.

TA feedback from the online PDs followed a similar pattern, with teachers expressing that the workshops went well overall. Many teachers were able to follow along with the demonstrations. Four TAs attending the Spring PD observed small technical issues with the network connection or screen sharing quality. However, these issues did not persist into the Summer PD. Furthermore, four TAs had also expressed high engagement and motivation within their breakout sessions and emphasized how the smaller group structure seemed to help teachers feel more comfortable asking questions and actively collaborating with each other and their TA. Two TAs noted that

Table 2: Summary of the Thematic Analysis of question 1 of teachers' feedback after Spring PD.

	Q1: What do you think about today's PD?	
Theme	**Count**	**Percentage**
unclear instruction	1	6.25%
need more time	5	31.25%
PD was beneficial	6	37.5%
step-by-step demos are preferred	2	12.5%
liked hands-on	1	6.25%
break-out sessions were beneficial	1	6.25%
Total	**16**	**100%**

some teachers did feel overwhelmed by the pace of the first session, which correlates to the teacher feedback regarding a need for more time to learn the concepts. The second PD in Summer had more consistent positive feedback from the TAs in attendance, where motivation and satisfaction among their groups were a common theme. Overall, teachers were reported to be very engaged, active, and motivated within their breakout sessions.

4 Reflection and Future Work

In this work, we presented the details of our approach to professional development in CT focused on transitioning teachers from a block-based programming environment to a text-based programming environment in Python. Our analyses of teachers' experience in the professional development shows that our approach was an effective training program to introduce and quickly prepare teachers for a school-wide adoption of Python. Our contribution is in providing curriculum resources in support of training teachers for CS and CT integration in their classrooms. The PD can be effectively conducted in a virtual modality. An ongoing challenge for teacher PD is the dedication of time that practitioners must devote to learning new concepts. In our future work, we are planning to extend our support to the teachers throughout the school year and provide more on-demand support for them. Furthermore, the plan is to support the middle school teachers and students through different service-learning programs within our University. We have been providing such service in the past with the block-based programming.

Table 3: Summer PD teacher survey score mean, median, and standard deviation (STDev) outcomes

ITEMS	**Mean**	**Median**	**STDev**
The facilitators were awesome	5	5	0
There was adequate time to cover everything	3.571	4	1.133
The activities were great	4.285	4	0.755
The atmosphere was engaging	4.285	4	0.755
The workshop made me feel like I could learn this over the summer	4.142	4	0.899
The content would be helpful in my classroom	4.142	4	1.069
My students could learn this material	4.428	4	0.534
I feel confident that I could eventually teach this material	4	4	0.816
My principal will be supportive of me implementing this next year	4.571	5	1.133
I was able to network and collaborate with my colleagues	4.571	5	0.786
I would recommend this session/ workshop to colleagues	4.714	5	0.487
I feel confident using Python	3	3	1.154
I feel confident writing code	2.857	3	1.345
I know how to teach Python concepts effectively	2.571	3	1.272
I can promote a positive attitude toward Python in my classroom	4	5	1.527
I can guide students application of Python concepts and tools while exploring other topics	3	4	1.632
I feel confident using computing tools as instructional tools within my classroom.	3.571	4	1.618
I can adapt lesson plans to incorporate Python	3.571	4	1.397
I can identify how Python concepts relate to Common Core Standards	4.285	4	0.755

References

[1] T. Bell, I. H. Witten, and M. Fellows, "Computer science unplugged," 2002.

[2] N. R. Council *et al.*, *Report of a workshop on the pedagogical aspects of computational thinking*. National Academies Press, 2011.

[3] J. M. Wing, "Computational thinking," *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, 2006.

[4] L. Blum and T. J. Cortina, "Cs4hs: an outreach program for high school cs teachers," *ACM SIGCSE Bulletin*, vol. 39, no. 1, pp. 19–23, 2007.

[5] C. Mellon, "Carnegie mellon center for computational thinking," *Retrieved December*, vol. 11, p. 2016, 2014.

[6] V. Barr and C. Stephenson, "Bringing computational thinking to k-12: what is involved and what is the role of the computer science education community?" *Acm Inroads*, vol. 2, no. 1, pp. 48–54, 2011.

[7] M. U. Bers, L. Flannery, E. R. Kazakoff, and A. Sullivan, "Computational thinking and tinkering: Exploration of an early childhood robotics curriculum," *Computers & Education*, vol. 72, pp. 145–157, 2014.

[8] M. U. Bers, I. Ponte, C. Juelich, A. Viera, and J. Schenker, "Teachers as designers: Integrating robotics in early childhood education," *Information technology in childhood education annual*, vol. 2002, no. 1, pp. 123–145, 2002.

[9] L. Darling-Hammond, M. E. Hyler, M. Gardner *et al.*, "Effective teacher professional development," 2017.

[10] A. Hargreaves and M. Fullan, *Professional capital: Transformng teaching in every school*. Teachers College Press, 2015.

[11] L. Matherson and T. M. Windle, "What do teachers want from their professional development? four emerging themes." *Delta Kappa Gamma Bulletin*, vol. 83, no. 3, 2017.

[12] R. Jocius, D. Joshi, Y. Dong, R. Robinson, V. Cateté, T. Barnes, J. Albert, A. Andrews, and N. Lytle, "Code, connect, create: The 3c professional development model to support computational thinking infusion," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 2020, pp. 971–977.

[13] H. Lee, M. Longhurst, and T. Campbell, "Teacher learning in technology professional development and its impact on student achievement in science," *International Journal of Science Education*, vol. 39, no. 10, pp. 1282–1303, 2017.

[14] M. Bower, L. Wood, J. Lai, C. Howe, R. Lister, R. Mason, K. Highfield, and J. Veal, "Improving the computational thinking pedagogical capabilities of school teachers," *Australian Journal of Teacher Education*, vol. 42, 04 2017.

[15] S.-C. Kong, M. Lai, and D. Sun, "Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy," *Computers & Education*, p. 103872, 2020.

[16] M. R. Nelson, "Focusing on teacher needs in k–12 cs education," *Commun. ACM*, vol. 59, no. 1, p. 5, Dec. 2015. [Online]. Available: https://doi.org/10.1145/2851202

[17] I. Z. Bargury, B. Haberman, A. Cohen, O. Muller, D. Zohar, D. Levy, and R. Hotoveli, "Implementing a new computer science curriculum for middle school in israel," pp. 1–6, 2012.

[18] R. Faherty, K. Nolan, and K. Quille, "A collaborative online micro: Bit k-12 teacher pd workshop," in *Proceedings of the 2020 ACM Conference on International Computing Education Research*, ser. ICER '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 307. [Online]. Available: https://doi.org/10.1145/3372782.3408113

[19] J. Rosato, C. Lucarelli, C. Beckworth, and R. Morelli, "A comparison of online and hybrid professional development for cs principles teachers," in *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 140–145. [Online]. Available: https://doi.org/10.1145/3059009.3059060

[20] D. Weintrop and U. Wilensky, "To block or not to block, that is the question: Students' perceptions of blocks-based programming," ser. IDC '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 199–208. [Online]. Available: https://doi.org/10.1145/2771839.2771860

[21] M. Mladenović, I. Boljat, and [U+FFFD] Žanko, "Comparing loops misconceptions in block-based and text-based programming languages at the k-12 level," *Education and Information Technologies*, vol. 23, pp. 1483–1500, 07 2018.

[22] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," in *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 14–18. [Online]. Available: https://doi.org/10.1145/1067445.1067453

[23] M. McCracken, V. Almstrum, D. Diaz, M. Guzdial, D. Hagan, Y. B.-D. Kolikant, C. Laxer, L. Thomas, I. Utting, and T. Wilusz, "A multinational, multi-institutional study of assessment of programming skills of first-year cs students," *ACM SIGCSE Bulletin*, vol. 33, no. 4, pp. 125–140, December 2001. [Online]. Available: https://kar.kent.ac.uk/13514/

[24] J. Piaget, *The Origins of Intelligence In Children*, 01 1952.

[25] M. Prensky, "Digital natives, digital immigrants part 1," *On the Horizon*, vol. 9, pp. 1–6, 09 2001.

[26] T. Clegg and J. Kolodner, "Bricoleurs and planners engaging in scientific reasoning: a tale of two groups in one learning community." *Research and Practice in Technology Enhanced Learning*, vol. 2, pp. 239–265, 11 2007.

[27] S. Turkle and S. Papert, "Epistemological pluralism and the revaluation of the concrete." 1992.

[28] A. Strauss and J. Corbin, *Basics of qualitative research techniques*. Sage publications Thousand Oaks, CA, 1998.