Building Collaboration and Securing Interest in Computer Science Education through Outreach Opportunities

Ms. Shaya Wolf, University of Wyoming

Shaya Wolf (swolf4@uwyo.edu) is currently pursuing her Ph.D. at the University of Wyoming. After completing Bachelors degrees in Math and Computer Science, her research focused on distributed systems, swarm communications, and encryption mechanisms. She is currently working on distributed continuous authentication systems and consensus protocols.

Fiona P. Moss, University of Wyoming

Fiona P. Moss is a Computer Science Master's student at the University of Wyoming. She completed her Bachelors in Computer Science in India. Her research interests are machine learning and cyber-security, and she is currently working on her thesis involving the application of machine learning techniques for Network Intrusion Detection.

Rasana Manandhar, University of Wyoming

Currently a Master's student at the University of Wyoming working on Continuous Authentication Systems.

Madison Cooley, University of Wyoming

Madison Cooley is currently an undergraduate computer science student at the University of Wyoming.

Rafer Cooley, University of Wyoming

Rafer Cooley is pursuing a Ph.D. at the University of Wyoming focusing on secure distributed systems. His interests include bio-inspired algorithms, Complex Adaptive Systems, and network protocols. He is currently working on bio-inspired solutions for securing industrial control systems.

Dr. Andrea Carneal Burrows, University of Wyoming

Andrea C. Burrows is currently an Associate Professor in the Department of Secondary Education at the University of Wyoming, where she teaches courses in science methods and pedagogy. Dr. Burrows taught at Northern Kentucky University for five years. In 2010, she was hired as an external evaluator to conduct research on community/university partnership relations at the University of Cincinnati. She has received several awards including the: 1) Lillian C. Sherman Award for outstanding academic achievement (2011); 2) UW College of Education outstanding research award (2015); and 3) UW College of Education outstanding service award (2016). Her research interests include partnerships with in pre-service and in-service teachers in STEM Education with a focus on engineering education applications. An active member of AERA, ASEE, ASTE, NARST, and NSTA, Dr. Burrows has presented at over 50 conferences, published in ranked journals (e.g. Journal of Chemical Education), reviewed conference proposals (e.g ASEE, AERA), and co-edits the CITE-Science journal. Additionally, she taught high school and middle school science for twelve years in Florida and Virginia, and she was the learning resource specialist for the technology demonstration school in Florida.

Dr. Mike Borowczak, University of Wyoming

Dr. Mike Borowczak is an Assitant Professor of Computer Science and the Director of the Cybersecurity Education and Research center (CEDAR) at the University of Wyoming. He earned his Ph.D. in Computer Science and Engineering (2013) as well as his BS in Computer Engineering (2007) from the University of Cincinnati. His research focused on detection and prevention of information leakage from hardware side channels. His current research interests include investigating the safety, resilience, and security of decentralized components, devices, and system architectures from theoretical modeling, to simulation and

practical implementations. He is also involved in K-20 CS/cybersecurity education research and was the 2019 RMS ASEE conference co-chair.

Mike also has over a decade of industry and research experience – mostly revolving around the semiconductor and bioinformatics industries – with specific experience at Texas Instruments, Intel, and Cincinnati Children's Hospital Medical Center. In addition to his industry experience, Mike spent two years, while completing his Ph.D., as a National Science Foundation GK-12 fellow – teaching and bringing real-world STEM applications in two urban high schools. Since then, he has worked with university faculty to promote and extend K20 STEM outreach in Ohio, Oregon, Texas, and Wyoming. He has authored peer-reviewed articles and papers, presented at national and international conferences, and taught undergraduate/graduate courses in Computer Security, Data Mining, VLSI and pedagogy in STEM.

Building Collaboration and Securing Interest in Computer Science Education through Outreach Opportunities

1 Abstract

Automation and mechanization require students to master the utilization and creation of new technology. Vital for potential careers, tomorrow's professionals require technological understanding to remain competitive in a job market driven by engineering advances. Regardless, most K-12 schools in the US and other countries do not currently offer Computer Science courses. To address these issues, multiple week-long summer outreach camps were designed for K-12 students and teachers. These camps delivered programming foundations in JavaScript and Python programming using online editors with both traditional and block-style programming options. Additionally, the camps utilized BBC Micro:Bits, pocket-sized computers fit with numerous functionalities. During interactive labs, participants collaborated with experts to develop various programs focused on topics such as password cracking, secure wireless communication, hardware hacking, securing Internet of Things devices, robotics, data collection, algorithm design, automation, decomposition, and pattern recognition. In addition to the explorative labs, students and teachers also worked with camp instructors in daily work sessions, using modified peer-instruction methods, to identify and analyze key Cybersecurity concepts such as modularization, simplicity, minimization, domain separation, least privilege, information hiding, layering, process isolation, resource encapsulation, and abstraction. The camp culminated with student demonstrations for their friends, family, and instructors.

Introduction

Reliance on immersive technology fosters the necessity to understand not just how to use technology, but how to create it. Vital for potential careers and protecting personal data, tomorrow's professionals require technological understanding to remain competitive in a job market driven by advancing technology. The Bureau of Labor Statistics projects that occupations in information technology and Computer Science will grow 13 percent from 2016 to 2026 [1]. Regardless, most K-12 schools in the US and other countries do not currently offer Computer Science courses, so state legislatures have turned their attention to incorporating these programming skills in K-12 curricula. For example, in the Wyoming Legislature's 2018 Budget Session, Senate File 29 was passed, adding Computer Science and computational thinking to the state educational program [2]. This bill requires Computer Science to be incorporated into the K-12 curriculum by the 2022-23 school year. However, due to a lack of emphasis on Computer Science, current instructors are not adequately equipped to teach such courses. Creating engaging lesson plans requires a comprehensive understanding of Computer Science topics. Crucial to the success of legislative efforts like SF29, training K-12 teachers to understand these concepts and teach them effectively necessitates appropriate outreach from experienced institutions.

Given the widespread use of technology, students have a basic understanding of Computer Science, but need refined programming skills to leverage this technology in their future professions. Waiting for higher education to expose students to these concepts inhibits their potential and stunts their academic growth. Our summer outreach camps combat the delay in instruction by introducing students to Computer Science early and exploring important concepts. Additionally, placing an emphasis on

diverse classes promotes inclusive future workplaces. Current tech industries suffer from a lack of diversity and introducing K-12 students to Computer Science early could be a potential solution. Lastly, since we store all of our personal data digitally (medical records, bank account information, private emails, messages, etc.), teaching cybersecurity to students from a young age equips them with the skills needed to protect their personal information early. Maintaining full attendance throughout each session, these outreach opportunities created a basis for students to work together on engineering solutions and for teachers to collaborate on teaching methods.

Prior Work

Workshops and extra-curricular activities have been proven successful in teaching Computer Science. During a round-table discussion on how Computer Science is taught in universities, six senior educators agreed that the brightest Computer Science students are passionate and learn through hands-on extra-curricular activities [3]. Additionally, outreach activities add intrinsic motivation for continuing learning in STEM fields. A recent study found that students responded better to learning environments that were personally relevant to them and almost all of the 729 participants preferred workshops over projects and lectures [4]. These findings suggest that students benefit from outreach activities with an out-of-school aspect. Our camps kept students interested by utilizing material that was relevant to them as well as engaging students in workshops with hands-on activities. We also focused on teacher development, encouraging teachers to expand their lessons to include fresh Computer Science topics. Successful STEM outreach programs demonstrate the importance of STEM in motivational ways, accented by teacher involvement and professional development, making support from schools and teachers crucial [5].

Computer Science education requires new approaches to STEM education in K-12 contexts. Facing low retention rates, STEM education must overcome teaching challenges to match the increase in available Computer Science jobs. Making a positive impact on retention begins before students reach college. Student environments affect retention rates and low-quality teaching result in weak learning and unsatisfied students who eventually leave the field to pursue other interests [6]. A recent report from the National Academics of Science for the National Science Foundation dictates that a lack of student support in the form of strong instruction and engagement in learning causes students to leave STEM related fields [7]. We find that instructors have the desire to teach new and exciting material that is relevant to their students, however realizing good ways to reformat lessons does not guarantee successful deployment in the classroom [8].

Prior work also shows that peer instruction can impact learning quality. In this classroom format, teachers leverage small group discussions to answer key topic questions. Peer instruction proves its impact through making students vital to the class by requiring their participation. By asking questions before and after peer instruction techniques were employed, a recent study showed that average correctness was increased from 44-47% to 63-68%, with a normalized gain of 35-41% [9]. In another study focused on overall classroom grades, they found that students in a class using peer instruction scored on average 5.7% better on their final exam that students taking the same course taught using traditional lectures [10]. In a third study focused on cybersecurity education, peer instruction was shown to improve student grades by 6%. These results were developed by looking at specific courses, finding that student failure rates were reduced by 61%. Additionally, 92% of students reported that discussions with peers during lecture helped them understand the material [11]. Lastly, peer instruction techniques aren't only applicable in universities. Positive results in university classes has lead to the adaptation of peer instruction in larger high schools and colleges. Additionally, research done into peer instruction in smaller liberal arts colleges shows that students benefit in similar ways [12]. This demonstrates promise and expresses viability for peer instruction in K-12 schools.

Methodology

Teaching Approaches

We employed many different ways to keep students engaged. First, we split students into two rooms based off of where they reported their programming skills. Students who considered themselves advanced where placed in one room and students who considered themselves beginners where placed in another. All of the students received the same tasks and levels of help, but formed groups with other students who had similar programming experience to their own. Additionally, students were free to move to a different room if they weren't engaged in their original room choice. Teachers, on the other hand, were split into different rooms based on what programming language they preferred to work in. This allowed participants to work with others with similar backgrounds and keep them engaged.

Technology Utilization

We utilized physical and virtual technology to keep the labs interesting. Physical components included Micro:Bits, badges, electronic lock-pick stations, and Micro:Bots, that engaged students by giving them the opportunity to learn through hands-on activities. We also used virtual components, such as the programming editors and optional online instructions, to aid in student comprehension. The inclusion of physical technology was vital in these labs because they engaged students and held their interest.

Micro:Bits, the basic building blocks for all of our labs, are pocket-sized computers that can be programmed and reprogrammed easily. They are fit with an array of sensors, including Bluetooth capabilities, programmable pins and buttons, an on-board compass and accelerometer, a temperature sensor and light sensor, radio features, and 25 programmable LED lights. We chose to use Micro:Bits because they are easy to program, low-cost, and can achieve exceedingly interesting goals. The Micro:Bit editors allow for students to bounce back and forth between block-style programming and text-based JavaScript programming. Another editor was also provided for students and teachers who wished to program in Python. This allowed for flexibility in the labs and provided participants with implementation options to suit their needs.

Each participant received a Micro:Bit badge. These badges resemble conference badges, but used a Micro:Bit to track learning progress through passing levels shown using the LED lights. The badges were programmed such that we could set the level by entering in a secret, random, and changing sequence of buttons. This was a way to incentivize students completing each challenge and lab.

We also created Micro:Bit-powered lock boxes. These were simple boxes to simulate smart locks that can be picked/hacked through discovering lab solutions. These stations were built such that students could attempt to pick different levels of locks, working their way up from a beginner lock to advanced locks. This gave students a firm and tangible goal to program towards. By showing a green sign when the lock was successfully unlocked and a red sign when the program failed, students could easily tell if their code worked or needed adjusting. Further, students could open the boxes and examine the Micro:Bit inside. By giving different inputs to the Micro:Bit, they could infer and learn for themselves what the lock was trying to do.

Each participant also programmed their own Micro:Bot, a small robot with two wheels and three sensors. These are simple bots that run with the help of a Moto:Bit, an extension board for the Micro:Bit that runs the motors for the wheels. Because these are simple bots, they were easy to assemble and fun to program. This captured student interest and provided physical technology to test out, rather than compiling and running programs entirely on their computer.

Instructional Strategies

These labs can be implemented in various classroom settings. They provide ideas for teachers as well as flexible applications. Each lab includes ways of tuning the complexity and difficulty of each activity, depending on the students being taught and the lesson time frame. We focused heavily on creating

labs that could be easily integrated into current curriculum and supplement existing lesson plans. This makes teaching Computer Science principles less demanding and much more attainable.

Participants were more engaged when labs started slow and worked up incrementally to more complex ideas. Before starting the labs, we introduced the participants to the Micro:Bits, then acquainted them with simple programming, and finally taught them necessary lab skills. Lab skills were taught through rudimentary activities. For example, to teach participants radio signal basics, participants worked in groups to send a message over radio to a pre-programmed Micro:Bit receiver at the front of the room. After discussing different methods for sending messages, groups were asked to receive a message from the Micro:Bit at the front of the room. After groups accomplished this, we again brought everyone together to talk about programming solutions. These simple tasks introduced the technology as well as radio signal basics in a simple and fun way.

Lab 01 - Beacons and Scanners

Lab Description

In the first lab, students used Micro:Bits as beacons and receivers to play a technology-based game of hide and seek. The objective of this lab was to search for and count how many beacons (transmitters) were hidden in a given area using their own programmed receivers.

Participants were given base-code (partially completed but functional code to facilitate/ease learning) that simply received messages from a beacon and were asked to test out it's capabilities. In minimal time, participants understood basic receivers and were asked to look for hidden signals given out by beacons. When they were close to a beacon, the display on their Micro:Bit would show a check-mark. After they tested the basic receiver, they looked at the code to see how the Micro:Bit worked. By giving students a working receiver to test first, and then showing them the code behind the scenes, it was much easier to build student understanding. Next, we made the field much larger, so receivers needed to be smarter. Participants extended the base-code such that the Micro:Bits displayed the strength of a received signal. This could be done in a lot of different ways, as long as it made sense to the participant. This way, participants could navigate around a room and tell whether or not they were following a beacon. Learning was accelerated by extending base-code, rather than forcing students to start from scratch. This eliminated the problem students generally have when learning new material; they don't know how to start.

Then, we placed beacons on different radio groups. Micro:Bits provide a primitive level of security while communicating data through radio waves by enabling users to switch through groups/channels. Much like a baby monitor or walkie-talkie, if a Micro:Bit sends out data on group one, another Micro:Bit on group two cannot receive it. The students extended their code again to allow their receiver to swap channels based on button pushes. Some participants even went one step further and created receivers that scanned through the different radio groups to find different beacons without having to press buttons on the Micro:Bit. Other participants instituted counters that kept track of how many beacons they had found.

Student Reception

Students were very competitive when they started out with the standard receivers aiming to find the most beacons. From this process, there were typically three major types of receivers implemented on the Micro:Bits: standard receivers, tunable receivers, and scanners. Standard receivers were built my less-experienced students and looked for beacons on one single group; looking on other groups required students to reprogram their Micro:Bit each time. Other students were able to implement tunable receivers giving them the liberty to switch through different groups and look for signals without having to reprogram their Micro:Bit. Advanced students executed scanner receivers that scanned through all the groups and counted all the beacons on each group. The array of potential solutions gave students of all experience levels a challenge they could get excited about and motivated complete solutions.

Teacher Reception

Starting out with the standard receivers, teachers walked about the rooms and hallways looking for signals, expressing delight when they found one. Unfortunately, very few teachers programmed tunable receivers and scanners because they struggled with understanding the implementation of the standard receiver and Micro:Bit functionality. Unlike the students, the teachers were uncomfortable moving forward without first understanding every component of the standard receiver, demonstrating less of a 'learn as you go' mentality and more rigidity. To remain flexible in our camps, we switched gears with the teachers and focused on how to get flexible labs to fit in with their current curriculum. We asked them how they could make labs like this better and how could this work in their classrooms.

Teaching Implications

This lab introduced the concept of secure communication. Treating this lab like a game, by sending group names around through the beacons to the receivers, hinted that a spark of healthy competition promoted even more interest and enthusiasm. Further, due to the number of extensions made to this lab, teachers could implement different parts of the lab to fit their needs. For instance, an elementary-level instructor may not need to implement a tunable or scanner receiver to teach students about secure communications. However, a secondary-level instructor may find the scanner receiver portion helpful in teaching how to think like an adversary and communication interception.

Lab 02 - Lock-Pick

Lab Description

Smart locks are becoming more popular in many homes and cars.These advancements however, require high levels of cybersecurity. This lab encouraged students to think like an adversary and try to crack smart lock simulations as well as demonstrated the importance of information hiding.

Given a lock box, students were tasked with opening the box by guessing the correct secret key. The lock boxes were constructed from cardboard, a Micro:Bit controller, and a Servo motor. The motor acted as the lock, which moved a flag to show either a red or green sign, depending on whether or not the secret key was guessed successfully. This motor was connected to a Micro:Bit that acted as a controller. The lock box controller accepted a binary string representing a number as the secret key.

In order to open the lock box, a collection of zeros and ones were sent from the students Micro:Bit key/lock pick to the lock box controller. Students achieved this goal by sending electronic keys over the radio from their Micro:Bit lock pick to the controller Micro:Bit in the lock box. After the students programmed a potential solution, they tested their solution on the lock box. The controller received signals from the lock pick and determined whether or not the secret key was guessed. If the lock box opened (a green flag is shown), they had guessed the correct key and were encouraged to work on a harder lock box with a more complex key. Many different levels of boxes were constructed, ranging in difficulty. This gave students easy starting points as well as complex goals to strive towards. Students were able to see the program that the controller was running for the easier boxes, but not with the more advanced boxes. This encouraged information hiding by showing students that hiding crucial information from prying eyes can greatly increase the difficulty of potential hacks.

Student Reception

Students enjoyed working with the lock boxes. Once they had cracked the first challenge, the rest of challenges seemed less daunting. Support was given to the students in the form of helpful hints given throughout the lab session. Students who seemed to be struggling were encouraged to work with students struggling with a different component. Some groups struck up some friendly competition and found it entertaining to see who could pick a given lock box first.

Teaching Implications

This lab simulates how electronic keys work, similar to office building key cards or smart locks. Since they had seen radio signals in the previous lab, they were able to extend this knowledge while learning about new technology, namely servo motor basics and smart lock simulators. Due to the range of difficulty in different lock boxes, students were able to hack some boxes, while finding it much more challenging to hack others. This helps to make this lab more flexible, as some lock boxes can be removed in the interest of time.

Lab 03 - Micro:Bots

Lab Description

This lab was conducted to acquaint students with the fundamental principles of communication interception/jamming through a practical Micro:Bit activity. Recall that in order to send and receive messages, both the transmitter and the receiver Micro:Bits must be on the same group/channel. In this lab, students programmed a controller and a Micro:bot. The Micro:bot is a robot with an embedded Micro:Bit that receives messages/directions being transmitted from the controller. Pressing buttons on the controller dictated the direction in which the user intended the micro:bot to move and transmitted those directions to the micro:bot, which then moved accordingly.

Initially, students were given base-code to help them understand how Micro:Bits communicate with each other through radio signals. The base-code for this lab consisted of two code files, one for the controller and the other for the Micro:Bot. The base-code set radio channels such that when button A was pressed, the radio group was decremented by 1 and when B was pressed, the radio group was incremented by 1. When buttons A and B are pressed together, the radio channel was locked in and the students could start transmitting instructions/directions to their bot. Further, the base-code for the controller was set up such that if the user presses button A, the message "left" was transmitted and if the user presses button B, the message "right" was transmitted. The first time the A and B buttons are pressed together, the message "forward" was transmitted to the Micro:Bot to tell the bot to start moving. This then toggled the code such that the next time the A and B button were pressed together, the message "stop" was transmitted to the Micro:Bot to stop it from moving. The base-code for the Micro:Bot enabled students to try out the bots and see how they worked.

As student began testing their Micro:Bots, they noticed that their bots were acting sporadically. As they feverishly looked through their code to find errors, it became apparent to camp leaders that all of the students were on the same radio channel. Given a couple hints and having them think back to the Beacons and Scanners lab, students began to realize that their bots were moving according to the directions being sent out by everyone's controllers, opening a discussion on radio jamming. Once students had their bots up and running (on their own secret group), students were tasked with controlling their bots through mazes. These mazes were drawn on the ground with masking tape and required the students to have fine-tuned control of their robot. After successfully completing different mazes, students were encouraged to build their robot code and add additional functionality such as speed control and reversing the Micro:Bot.

Student Reception

The students enjoyed this lab and it's hands-on approach to cybersecurity principles. They were excited about operating the Micro:Bots and quickly figured out ways to operate them. Then they went on to add more functionality to the code to complete the mazes faster than other groups/students. They used many different features of the Micro:Bit such as shake and tilt to accomplish these upgrades. They attempted hacking each other's Micro:Bots and from that knowledge, built defenses for their own bots. This was done through sanctioned hacking times where students could test their hacking attempts and other students knew to be ready with their improved defenses.

Teaching Implications

This lab introduced communication security concepts vital to many wireless communication systems. Engaging student participation in such lab activities sparked their interest in Computer Science. More specifically, this activity helped students understand cybersecurity concepts such as defense in depth, keeping it simple, and thinking like an adversary. Teachers could also design a task in which students could be taught how to cipher the messages being transmitted to prevent hacking thereby encouraging them to learn about confidentiality as well as other cybersecurity concepts.

Lab 04 (Backup) - Fitness Tracker

Lab Description

This lab was a backup in the event that one of the other labs ran into technical difficulties. In this lab, participants program their Micro:Bits to mimic Bluetooth fitness trackers. They program their Micro:Bits to gather data using the on-board accelerometer and then graph the data to determine different activities (walking, running, jumping, and so on). Finally, they intercept a partner's data and infer their actions based off of their graphs. This lab allows the students to learn important Computer Science and cybersecurity principles in manageable chunks. We also provide all of the necessary graphing components, making the lab suitable for younger groups who haven't seen algebra in their math classes yet. For older groups, a conversation about data interpretation allows for a discussion on data mining basics. Students also learn data collection techniques as well as data representation. This is done through programming, pattern recognition, and group work.

Student Demonstrations

Our camps concluded with student demonstrations of their work throughout the week. Students were given a chance to pick their favorite lab, perfect their code from earlier in the week, and show off their favorite accomplishment. This gave students a chance to talk with others about the different things that they had learned throughout the week. In addition to presenting to their peers, students demonstrated their learning to their other friends and family. Even further, we had the teachers walk through and see the student presentations as well. This showed teachers the different teaching avenues for Computer Science and what students enjoyed most from their time at camp. Personalized lab demonstrations by most students Students demonstrated their work in many different ways. Some students chose to work on their code in a group and demonstrate as a group. Other students chose to work in a group but demonstrate on their own. Additionally, some students were more interested in working in groups than they were demonstrating their favorite lab, choosing to demonstrate less-preferred work in order to work with others. In addition to these trends, we saw older students that preferred to work on their own projects and demonstrate those instead of labs accomplished through the week. Other older students chose to work in groups with younger students to help them finish their projects. This showed higher level mastery of the same lessons while receiving similar instruction.

Conclusion

This free camp enabled us to make Computer Science more accessible and enjoyable among K-12 students and teachers. Participants learned concepts of Computer Science and cybersecurity through interactive Micro:Bit activities and also learned to code in Python and block-style JavaScript. Students demonstrated their learning from the camps to their family and friends. Teachers found ways to incorporate Computer Science lab activities in their curriculum. Participants were able to keep their badges, including the Micro:Bit, so they could use them in the future. We hope to improve future camps and continue helping students and teachers master Computer Science learning.

Acknowledgments

This work was supported by 1) The National Science Foundation (NSF) and National Security Agency (NSA) GenCyber Award #H98230-18-1-0095 (called GenCyber:COWPOKES); 2) The NSF Noyce Grant No 1339853 (called SWARMS); and 3) The US federal Math and Science Partnership grant under No Child Left Behind (NCLB) (P.L.107F110, Title II, Part B) administered by the Wyoming Department of Education MSP Grant No. 1601506MSPA2 (called RAMPED). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF, the NSA, or the U.S. government.

This work was completed with the help of students from the University of Wyoming through the direction of the College of Engineering's CEDAR (Cybersecurity EDucation And Research) Center and the College of Education's SEED (Science and Engineering Education Development) Center.

References

[1] Computer and information technology occupations. *U.S. Bureau of Labor Statistics*, April 2018.

[2] State of Wyoming. Senate file no. sf0029, 2018.

[3] J. Voas, R. Kuhn, C. Paulsen, and K. Schaffer. Computer science education in 2018. *IT Professional*, 20(1):9–14, January 2018.

[4] Johanna Vennix, Perry den Brok, and Ruurd Taconis. Do outreach activities in secondary stem education motivate students and improve their attitudes towards stem? *International Journal of Science Education*, 40(11):1263–1283, 2018.

[5] Farzana Aslam, Arinola Adefila, and Yamuna Bagiya. Stem outreach activities: an approach to teachers' professional development. *Journal of Education for Teaching*, 44(1):58–70, 2018.

[6] Michail N. Giannakos, Ilias O. Pappas, Letizia Jaccheri, and Demetrios G. Sampson. Understanding student retention in computer science education: The role of environment, gains, barriers and usefulness. *Education and Information Technologies*, 22:2365–2382, September 2017 2017.

[7] Peter Gwynne. New approaches to stem education. *Research-Technology Management*, 61(2):6+, March-April 2018.

[8] Tamara D. Holmlund, Kristin Lesseig, and David Slavit. Making sense of "stem education" in k-12 contexts. *International Journal of STEM Education*, 5(1):32, Aug 2018.

[9] Beth Simon, Michael Kohanfars, Jeff Lee, Karen Tamayo, and Quintin Cutts. Experience report: Peer instruction in introductory computing. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, SIGCSE '10, pages 341–345, New York, NY, USA, 2010. ACM.

[10] Jaime Spacco, Julian Parris, and Beth Simon. How we teach impacts student learning: Peer instruction vs. lecture in cs0. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, SIGCSE '13, pages 41–46, New York, NY, USA, 2013. ACM.

[11] I. Ahmed and V. Roussev. Peer instruction teaching methodology for cybersecurity education. *IEEE Security Privacy*, 16(4):88–91, July 2018.

[12] Leo Porter, Saturnino Garcia, John Glick, Andrew Matusiewicz, and Cynthia Taylor. Peer instruction in computer science at small liberal arts colleges. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '13, pages 129–134, New York, NY, USA, 2013. ACM.