Improving Computer Science Education through CSEd Week Workshops

Matthew F. Tennyson, Sarah G. Fields, Ethan J. Hill
Department of Computer Science & Information Systems, Bradley University, USA

Abstract

Computing is pervasive. It permeates our lives in countless ways on a daily basis. As such, computer science should be an essential component of a well-rounded education. Unfortunately, the state of computer science education is poor and there are indications that it is getting worse. Computer Science Education Week was established in 2009 to raise awareness regarding the importance of computing in today's society and to recognize the need to bolster computer science education at all levels. In support of Computer Science Education Week, the Department of Computer Science and Information Systems at Bradley University has organized and facilitated several activities to help raise awareness and excite young people about computer science. Workshops in programming, Lego robotics, mobile computing, and mathematics were offered to local high school students. Through the successes and failures of our programs, valuable lessons and insights have been gained into improving computer science education outreach activities.

1. Introduction

As the level and abundance of technology around the globe continues to increase, the field of computer science is becoming increasingly vital. The U.S. Bureau of Labor Statistics predicts that more jobs will be created in the computing industry through 2018 than in any other STEM-related industry [1]. With the nearly constant exposure to technology in our daily lives, we might expect to see these job openings easily filled by aspiring computer science and information systems students. However, the number of undergraduate enrollments in computer science degree programs has actually decreased in recent years. As Figure 1 shows, this is expected to cause a gap in the number of job openings and the number of college graduates available to fill those positions [2].

Computer science is undeniably critical in today's society. Computers and computing are nearly ubiquitous. Computer science drives innovation and growth, it provides rewarding job opportunities, and it prepares students with the knowledge and skills necessary for a future filled with technology [3]. Even students without plans to enter the computing industry can make use of fundamental computing concepts. Education in computer science exposes students to critical thinking and problem solving, and it helps promote effective communication.

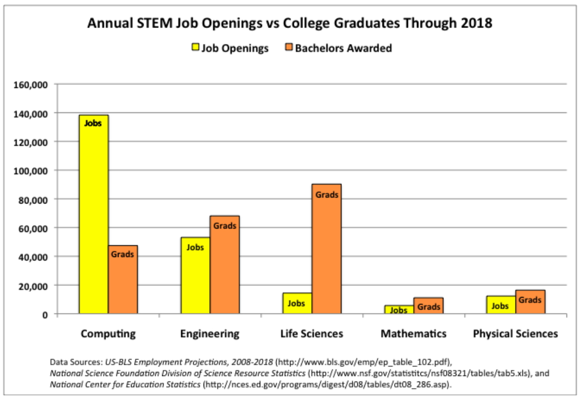

Figure 1. A comparison of the number of job openings to the number of college graduates available to fill those openings through 2018 [2].

2. Current State of CS Education

Unfortunately, computer science education in our school systems falls short of producing the necessary skilled workforce. According to a 2010 report published by the Association of Computing Machinery (ACM) and Computer Science Teachers Association (CSTA), "U.S. K-12 education has fallen woefully behind in preparing students with the fundamental computer science knowledge and skills they need for future success" [4]. The report goes on to claim that as the significance of computing has increased, that the quality of computer science education has decreased.

Only nine states recognize a computer science course as filling a graduation requirement, and no states require a course in the subject before graduation [5]. A report from the Computer Science Teachers Association (CSTA) indicates that the number of high schools in the United States included programming (generally considered a core component of computer science) in their curriculum dropped 13% from 2005 to 2007 [6]. Only eleven states rank above 80% compliance with the model curriculum developed by the CSTA, and virtually every state places higher emphasis on skills like programming than on substantial computer science concepts [5].

Many of the offered computer science courses in high schools are AP Computer Science, but even the allure of college credit isn't keeping enrollment up. A study in 2005 revealed that the number and diversity of students taking AP Computer Science courses appears to be dropping [7]. The number of students taking the A exam dropped 8% between 2002 and 2004, and the numbers of the AB exam dropped 19%.

There is also a disconcerting trend of dropping participation of women and minorities. In 2005, only 23% of students who took AP computer science courses were female. In 2007, the percentage dropped to 17% [6]. On the 2004 AB exam, females constituted only 11% of the takers, and under-represented minorities constituted only 6%.

Although the aforementioned studies specifically focus on computer science education in the United States, it appears the trend might also apply in other nations as well. For example, one study by the European Commission remarks, "In recent years, many studies have highlighted an alarming decline in young people's interest for key science studies and mathematics" [8]. Also, the Canadian Association of Computer Science (CACS) has embraced Computer Science Education Week, whose purpose includes recognizing the need to bolster computer science education at all levels [9].

It is imperative that computer science education be revitalized and that students be well-versed in computer science in order to succeed in today's society. To rouse in students an interest in computer science, educators must engage students starting from an early age. Introducing computing topics early in schools can ensure that students with an aptitude and interest are fostered appropriately. Offering extracurricular programs to young students also has great potential to increase interest.

3. Computer Science Education Week

The week of Grace Murray Hopper's birthday, December 9, is designated each year as Computer Science Education Week (CSEdWeek) [3]. Hopper was a pioneer of computer science, instrumental in many early developments in the field including the creation of the first compiler as well as the COBOL programming language. Hopper was known as an enthusiastic and vigorous teacher of math and computer science. So it's fitting that CSEdWeek – the purpose of which is to raise awareness of computing and its critical role in society and the consequential importance of computer science education – is dedicated in her honor. Throughout the week, parents, teachers, and policymakers across the world engage in activities to promote computer science and computer science education.

CSEdWeek is sponsored by the ACM, the CSTA, and several other professional, commercial, and service organizations. It is an activity of the Computing in the Core coalition [10]. Organizations and individuals across the globe participate by communicating about the importance of computer science and computer science education. Teachers at grade schools and high schools assign tasks aimed at raising awareness and inciting excitement for computer science in their students. Various forms of competitions and celebrations are held, and institutions of higher education execute outreach programs.

4. CSEdWeek at Bradley

The Department of Computer Science and Information Systems at Bradley University has actively and enthusiastically participated in CSEdWeek since its second year. In 2010, two area high schools were invited to participate in all-day events that included campus tours, hands-on computer science workshops, and lessons about careers and degrees in computer science. In 2011, the program was different in several ways. The workshops were scheduled in the evening as after-school activities, rather than during the day. A wider variety of workshops were offered, and they were open to students from any school. During both years, a video gaming competition was held that included both high school and university students.

There were many successful aspects of both the 2010 and 2011 programs. Indeed, there were also aspects of each program that were not so successful. This paper provides a comparison of both programs and an explanation of the major workshops in order to give insight and provide recommendations regarding the planning and implementation of similar activities in the future.

4.1. Overview of 2010 Activities

In 2010, the university hosted its first substantial CSEdWeek program. The program was planned over the course of several months by a team of graduate and undergraduate students as well as university faculty and experts. The university invited students from two local-area high schools to visit its campus. The activities spanned a total of three days and included campus tours, programming workshops, robotics workshops, lessons about careers in computer science, information about degrees in computer science, and a gaming competition.

The program spanned three days. On the first day, one high school was invited to campus for an all-day visit. During this visit, the participants were welcomed and given an introductory presentation that included an overview of computer science, careers in computer science, and degree programs in computer science. The students then toured campus. Next, the students attended one of two workshops: either a programming workshop or a robotics workshop. At the midpoint of each workshop, the students took a break for lunch, and then they returned to finish the second half of their respective workshop. At the completion of the workshops, the students were given a farewell and the day ended. A schedule is shown below:

8:00 a.m. Welcome
8:30 a.m. Introductory Presentation
9:15 a.m. Campus Tours
10:00 a.m. Workshops (first half)
11:30 a.m. Lunch
12:30 pm. Workshops (second half)
2:00 p.m. Farewell

The second day of the CSEdWeek program was merely a reprise of the first day for a different high school. The third day of the program was scheduled on a Saturday, and it consisted of an all-day gaming party. At the gaming party, several video game tournaments were held. Students from the high schools that visited campus earlier in the week returned to participate in the competitions. University students also participated.

The programming and robotics workshops were the academic focus of the overall program, and warrant additional discussion in the following sections. In the programming workshop, students learned basic programming concepts by creating simple games and simulations for the PC using the Greenfoot [11] development environment. In the robotics workshop, students learned basic robotics concepts by building and programming robots using Lego Mindstorms robotics kits. The workshops were held simultaneously, allowing each student to choose only one of the workshops to attend.

A survey of the high school participants was taken after the week had ended. The results were overwhelmingly positive. 22.2% of the participants had a positive impression of Bradley beforehand, while 73.7% had a positive impression of Bradley afterwards. 47.4% were considering a career in computer science before, while 57.9% were considering a career in CS after. 100% of participants said they would like to participate in similar activities in the future.

4.2. Programming with Greenfoot

The programming workshop aimed to get students experience in programming logic and syntax in a user-friendly environment. The programming environment Greenfoot, developed by Dr. Michael Kölling, allows younger programmers to learn the Java language while interacting with graphics. The workshop was based on Dr. Kölling’s crab scenario [11, 12]. Because the Greenfoot environment has been well documented, only a summary of the workshop is provided here. For additional discussion of the workshop, see “Computer Science Education Week Workshops for Secondary Students” [13].

The crab scenario had students work with classes and objects represented visually by pictures of crabs in a finite virtual space. Greenfoot’s development environment shows how manipulating methods in the crab class affect how individual crabs on the graphical display operate. The instructor of the workshop discussed key object-oriented concepts, and then helped students execute new code on the computers. The concepts were at an introductory level and focused on the crab scenario.

The first task for students was to get the crabs moving across the screen. Next, conditional statements were explored to affect movement patterns. Random number generation was added to create variation in the crab movements. Once the crabs were perfected, students created a new predacious lobster class based on their crab class. The lobsters would eat crabs, and a worm class was created as food for crabs. Finally, to make the example into a game for students, they learned how to read keyboard input to move the pieces around. At the end, students recorded their own sounds to put into the game.

4.3. Robotics with Lego Robots

The robotics workshop allowed students to program and interact with Lego Mindstorms robots. The Mindstorms manual contains instructions on how to assemble the basic robot [14], so the workshop focused on programming the movement of the base once it was assembled. The instructor of the workshop led the students through three objectives: assemble the robot, program simple movement, and add sensor input to the robots. For additional discussion of the workshop, see “Computer Science Education Week Workshops for Secondary Students” [13].

In the first section, the students built the Lego Mindstorms NXT model based on the instruction manual, which had picture instructions for assembling the Lego bricks and rods. The robot features an on-board computer hooked to motors that individually power tread wheels. The on-board computer has several pre-programmed, simple movement programs that students could execute to get used to their movement. Several students had previous experience with Mindstorms models and robotics, so they jumped quickly to the second or third step.

In the second section, the students began assembling their own programs in Lego’s development environment. The programs are represented in graphical form, but the control structures and linear flow resemble typical programming languages. Students could modify samples programs or create their own from scratch to accomplish simple tasks provided by the instructor, and students were encouraged to experiment with code. Challenges posed included moving forward, completing turns, and doing figure eights. Turning required the students to consider which treads needed to move in specific directions to accomplish a turn.

In the third section, the students added sensor attachments to the robot base. The Mindstorms kits include touch, color, and sound sensors, and the touch sensors were encouraged due to their simplicity and accuracy. Students were allowed to be creative, but specific challenges such as reversing or turning when bumped were posed. Loops and conditional statements were required for this section, so students grew familiar with programming logic.

The students worked in pairs due to limited supply of Mindstorms kits. The pairs also helped the students speed up the assembly process, and students could bounce ideas off their partner when programming the robot. The students who experimented with robot design or came up with their own challenges to tackle were particularly engaged.

4.4. Overview of 2011 Activities

In 2011, the university held its second annual CSEdWeek program. The structure of the event was changed from the previous year. The program still spanned three days; however, all of the activities were scheduled in the evening. The number of hands-on workshops was increased from two to four, while some of the other less-academic activities (such as the introductory presentation, campus tours, and lunch) were eliminated. The workshops still spanned approximately three hours, and a break in the middle provided snacks and drinks but no meal.

The previously detailed programming and robotics workshops were reprised from the previous year. Two additional workshops were added: one in mobile computing and another in mathematics, both of which are detailed in the sections below. In the mobile computing workshop, students learned to create applications for mobile devices using Google's Android App Inventor development environment. Students also learned about wireless sensor networks (WSNs) through interactive demonstrations. In the math workshop, students applied basic mathematics in interesting and engaging – but also practical – ways.

The programming and robotics workshops were held during the first evening, the mobile computing and math workshops were held during the second evening, and a gaming party was held during the third and final evening. The paired workshops were held concurrently, so students could only choose one of each two to attend. Several students chose to repeat a workshop they had previously attended.

4.5. Mobile Computing with Android Devices

The mobile workshop had students programming applications on the widely available platform of Android devices. As apps grow in popularity, the chance to create something useful showed the students the relevance of computer science in their lives. Google's Android Development Tools (ADT) and the Eclipse development environment together allowed students to both create and execute their programs [15]. The workshop was adapted from a tutorial developed by Lars Vogel [16]. The workshop was broken down into four sections: get used to Android, build text-based programs, build graphical programs, and deploy the programs on a mobile device. Each section began with a lecture and explanation from the instructor, and then students implemented the concepts in their individual programs. For additional discussion of the workshop, see "Computer Science Education Week Workshops for Secondary Students" [13].

In the first section, the students grew familiar with the Android platform and the intent of the workshop. The ADT has an Android emulator, so the students could interact with their programs without a physical device. The applications were written in Java, so the students were introduced to programming and Eclipse basics in order to help them write the programs.

In the second section, the students began building text-based programs. A hello world program was their first accomplishment. Once the students correctly displayed the text "Welcome to CSEdWeek," they were shown how to display user menus, read input, and work with variables. Each student wrote their own temperature converter that could read in Fahrenheit and convert it to Celsius or vice versa. They needed to store, manipulate, and print the temperature variables, so students learned valuable programming concepts.

In the third section, these programs were adapted into a graphical format. The instructor introduced the concept of GUI components, and then participants were guided through adding graphical displays to their programs. The hello world program was a simple display, but the interactive temperature converter used text fields, radio buttons, and more.

In the fourth section, these programs were deployed to an Android device. While in testing, the students only used the emulator to see their programs, but now students downloaded them onto devices provided by the facilitators. Several students had their own Android smartphones to launch the programs on, and these students seemed enthusiastic about seeing their work on their own devices. One student even declared, "I feel like I am a genius at Android programming." All of the students enjoyed seeing the Java code they developed in a useful application format.

4.6. Exploring Math through Cryptography and Games of Chance

The math workshop walked students through several math problems in the form of games of chance and explored basic cryptography. The fun approach to the subject also demonstrated to students that computer science is about more than just programming. The students were led through a discussion of the Monty Hall problem, the birthday paradox, analysis and discussion of which can be found in [17]. The students also learned about basic cryptography modular arithmetic. For additional discussion of the workshop, see "Computer Science Education Week Workshops for Secondary Students" [13].

In the first discussion, the instructor presented the Monty Hall Problem as the host of the game show *Let's Make a Deal*. In the show, one door onstage hides a prize such as a car, and two doors hide a goat. The participants are supposed to guess which door the car is behind in order to win. For the workshop, each student, playing a game show participant, is allowed to pick one of three "doors." One door conceals a prize like candy, and the other two contain nothing of value. Once the contestant has chosen, the host opens one unpicked door that does not contain the prize. The contestant must decide whether to stick with her current choice or choose the other closed door. By recording the results of each student's choice, the paradox can be shown that changing doors results in 1/2 chance of success while remaining with the first choice results in 1/3 chance of success. A computer simulation can also with a larger sample size the probabilities. The counter-intuitive nature of this problem allowed the students to discuss probability in an engaging and thoughtful manner.

In the second discussion, the students were presented with the birthday paradox: given only twenty-three people, the chance that any two of them share a birthday is greater than 50%. The chance that any number of people will have two who share a birthday can be calculated using combinatorial principles and probability. The counter-intuitive math principles once again allowed the students to engage in mathematical reasoning with a fun approach. The birthdays of the participants or a computer simulation can help demonstrate the empirical results of birthday probabilities.

In the third discussion, the basics of cryptography and modular arithmetic were introduced. Substitution cipher basics provided an entry point to the topic, and then students were asked to decrypt a message encrypted in Caesar cipher. The students were asked to encode their own message to exchange with another participant and decode. Next, the topic of modular arithmetic was introduced. The twelve-hour clock was used as an example of how modulus values affect additions of time exceeding twelve hours. This discussion led to the idea of public-key cryptography using RSA as an example. The exact mathematical principles underlying the RSA protocol were not discussed, but an example demonstrated to the participants the difficulty of factoring large numbers. The instructor explained that encoding a message m with public key n and exponent e requires the calculation of $c = m^e \pmod{n}$. One participant chose m in secret, and the rest of the students were asked to recover the message independently. The difficulty of this task even with small numbers illustrated the concepts of public-key cryptography.

4.7. Comparison

There were both positive and negative aspects of both the 2010 and 2011 events. Most of the comparisons can be made with respect to the following issues:

(1) Participation
(2) Logistics
(3) Expenses
(4) Topics covered

In 2010, only two high schools were involved. The schools and students viewed the event as a field trip. Garnering participation was relatively easy. In 2011, on the other hand, the events were open to anyone from any school. However, because the events were scheduled in the evening and not held as a school-sponsored activity, the participants had to invest their personal time and find their own transportation in order to attend. During the planning stage, the expansion of the targeted audience to an unlimited number of schools was viewed as a benefit, but in practice it ended up being a detriment. Participation dropped by about 50% from 2010 to 2011. However, the goal of diversifying the participants was achieved, considering all of the 2010 participants came from only two schools while the 2011 participants came from six different schools.

Logistically, the 2011 event was considerably simpler. Because the event was held in the evenings, it minimized a number of scheduling conflicts. It was more convenient for volunteers to facilitate the workshops. The elimination of an all-day itinerary that included tours, lunch, and a variety of other activities simplified the facilitation of the event greatly. However, the planning of the 2011 event was not without its logistical difficulties. Because specific schools were not targeted, the event had to be marketed and advertised. Also, because specific schools were not involved to inherently manage the number of participants, a registration system had to be employed.

Managing costs was a significant factor in planning both the 2010 and 2011 events. In 2010, due to the all-day field trip nature of the events, the costs were high. Also, because it was the inaugural year, there were intrinsic start-up costs. In 2011, on the other hand, a relatively small amount of money was spent. Resources from the previous year were reused, meals were not provided, and students provided their own transportation to the events. In 2010, only two hands-on workshops were facilitated, but other content was included (such as an overview of computer science, information about careers and degrees in computer science, and campus tours). Because the event lasted all day, time allowed for longer workshops. In 2011, more workshop topics were covered and offered on different days, which allowed for the participants to attend multiple workshops on whatever topics interested them.

5. Lessons Learned

Several lessons have been learned over the course of Bradley University's involvement in Computer Science Education Week. And, from these lessons, recommendations can be harvested in the hope and with the goal of enhancing similar outreach activities at other institutions and enhancing the educational experiences of students in general.

It is easier to garner participation when working with a small, targeted number of schools. A trade-off is that the participants will not be as diversified. However, in the context of an annual event, the diversification of the participants can be managed over the course of multiple years. Perhaps only one or two schools will be reached in the first year, but another one or two schools will be reached in each subsequent year. Diversification will be achieved over a span of time.

It is easier and more feasible for high schools and high school students to participate when the events are held during the day as part of a school-sponsored event. From the high school teacher's perspective, it's an opportunity to provide a valuable learning experience to her students during the course of a regular school day. From the student's perspective, it's a valuable learning opportunity outside her normal learning environment during the course of a regular school day. The activities are held at a time when the student and teacher are normally engaged and don't conflict with personal commitments.

It is difficult, but necessary, to find an optimal balance between activities that enrich the academic experience of the participant and those activities that enhance the overall experience of the participant. The primary mitigating factor is cost. The campus experience is intrinsically valuable to the high school student, but costly to the facilitator of the event in numerous ways.

6. Conclusion

As the computing industry continues to grow and technology continues to spread, we must foster an interest in computer science in our youth. We must ensure that computer science becomes an integral component of our primary and secondary curricula. CSEdWeek offers educators an opportunity and a platform to promote the importance of computer science and computer science education. Computer science should not be considered an elective or an unreachable career path to anyone.

Offering workshops and information sessions to youth is one good way to engage students. Colleges and universities are ideal hosts of such activities because young students will see (if only a glimpse of) what the university experience is all about. We believe the activities and workshops discussed herein provide valuable computer science exposure and skills to high school students. It is important for participants to be actively engaged while learning what computer science is all about.

7. Acknowledgements

Bradley University’s CSEdWeek activities were funded in part through a grant from the Office for Teaching Excellence and Faculty Development. A special word of thanks is extended to all the faculty members and students who spent many hours planning and facilitating the events.

8. References

[1] U.S. Bureau of Labor Statistics, *Occupational Outlook Handbook, 2010-11 Edition*, U.S. Department of Labor, Washington, DC, 2010.

[2] The Market for Computing Careers, http://cs.calvin.edu/p/ComputingCareersMarket. (Access date: 28 March 2012).

[3] Computer Science Education Week (CSEdWeek). http://csedweek.org. (Access date: 11 May 2012).

[4] C. Wilson, L.A. Sudol, C. Stephenson, and M. Stehlik, *Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age*, ACM/CSTA, New York, 2010.

[5] A. Tucker, F. Deek, J. Jones, D. McCowan, C. Stephenson, and A. Verno, *A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee*, CSTA, New York, 2004.

[6] J. Gal-Ezer and C. Stephenson, "The Current State of Computer Science in U.S. High Schools: A Report from Two National Surveys", *Journal for Computing Teachers*, ISTE, Spring 2009, pp. 1-5.

[7] C. Stephenson, J. Gal-Ezer, B. Haberman, B., and A. Verno, *The New Educational Imperative: Improving High School Computer Science Education*, ACM/CSTA, NewYork, 2006.

[8] M. Rocard, P. Csermely, D. Jorde, D. Lenzen, H. Walberg-Henriksson, and V. Hemmo, *Science Education Now: A Renewed Pedagogy for the Future of Europe*, European Commission, Brussels, 2007.

[9] Canadian Association of Computer Science (CACS). http://www.cacsaic.ca/. (Access date: 11 May 2012).

[10]Computing in the Core. http://computinginthecore.org/. (Access date: 11 May 2012).

[11] M. Kölling, *Introduction to Programming with Greenfoot*, Prentice Hall, Upper Saddle River, New Jersey, USA, 2010.

[12]M.Kölling, "The Greenfoot Programming Environment", *ACM Transactions on Computing Education*, 10 (4), 2010, pp. 1-21.

[13] M.F. Tennyson, S.G. Fields, and Y. Wang, "Computer Science Education Week Workshops for Secondary Students", *Journal of Computer Sciences in College*, Consortium for Computing Sciences in Colleges, October 2012, pp. 39-46.

[14] The Lego Group, *Lego Mindstorms User Guide*, The Lego Group, Slough, UK, 2009.

[15] Android Developers. http://developer.android.com/. (Access date: 28 March 2012).

[16]L.Vogel, Android Development Tutorial, http://www.vogella.de/articles/Android/article.html. (Access date: 28 March 2012).

[17]E. B. Burger and M. Starbird, *The Heart of Mathematics: An Invitation to Effective Thinking, 3rd Edition*, Wiley, New York, 2009.

