Reaching Out to Aid in Retention: Empowering Undergraduate Women

Rebekah Overdorf
Moravian College
1200 Main St.
Bethlehem, Pennsylvania 18018
strjo01@moravian.edu

Matthew Lang
Moravian College
1200 Main St.
Bethlehem, Pennsylvania 18018
lang@cs.moravian.edu

ABSTRACT

Creating programs that engage undergraduate women with the broader community and encourage them to take an active role in changing the underrepresentation of women in computer science can effectively address *both* retention *and* recruitment of women in the discipline.

This paper is an experience report describing the creation and outcomes of an outreach program for K–12 girls run entirely by undergraduate women.

The contributions of this paper are the description of the creation of a successful student-led outreach program and a set of active-learning modules for K–12 students that illustrate advanced topics.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education

General Terms

Human Factors

Keywords

women in computing, K–12 outreach, retention, gender issues

1. INTRODUCTION

The recruitment and retention of women in undergraduate computer science programs are long-standing problems in computer science education. Though the computer science education community has made correcting the underrepresentation of women in our discipline a priority, [14, 6, 18] little has changed over the past 20 years [13].

A frequently used recruiting tool has been for CS educators to run workshops or roadshows for K–12 students. These programs typically have the goal of dispelling misconceptions that young people have about computer science as well as inspiring students to choose computer science as a topic of study [8].

Our approach has been similar. Over the course of 2009–2010, we ran a series of short workshops and lectures for Girl Scout troupes, camps, and K–12 classrooms. However, instead of having these activities be faculty-designed and delivered, we built a program in which undergraduate women were given complete ownership of both the design and delivery of the workshops.

Recent research suggests that environmental cues that align with computer science stereotypes (*e.g.*, science fiction movie posters, comic book imagery, etc.) foster a feeling of not belonging among women considering computer science as a field of study [7]. So, the advantage of such an approach is two-fold: first, by placing undergraduate women in front of audiences of K–12 girls, the girls are presented with a positive and non-stereotypical image of a computer scientist. Second, the undergraduates involved gain a greater appreciation of computer science and a deeper connection to the discipline, which aids in retention.

In July of 2010, the Girl Scouts of Eastern Pennsylvania invited the authors to present at a summer day camp for K–12 girls. The 2.5 hour-long workshop was to be delivered to nearly 50 girls and to be focused on the theme of the camp—criminal justice.

The authors along with a group of undergraduate women designed a series of active-learning activities that served to illustrate fundamental ideas ranging from information theory to the separation between specification and implementation. These activities were received with near-unanimous approval by the girls and led to a dramatic shift in responses to a survey administered before and after the workshop.

Beyond the shift in attitudes among the girls involved, the experience led to a shift in attitude among the undergraduates who designed and administered the workshop. Beginning in April 2010, a group of undergraduate women met regularly with each other and a faculty mentor to design the workshop. Inside and outside of these meetings, the women involved researched a broad range of computer science topics.

Though there exists a wealth of pre-designed activities and lectures for K–12 students, the women chose to design their activities from scratch. Furthermore, despite the availability of programming languages accessible to K–12 students (*e.g.*, LOGO, Scratch, etc. [16, 19]), the women chose to design a series of "unplugged" [11] activities.

In order to align with the theme of the camp, the women researched topics that may have applications in criminal justice. Pursuit-evader games, the art gallery problem, steganography, cryptography, and DNA sequencing were researched and subsequently discussed with a faculty mentor in meetings. In the end, the women chose to illustrate ideas in information theory with an activity related to facial recognition; to discuss abstraction and generality by relating facial and fingerprint recognition to DNA fingerprinting; and to introduce specification, algorithms, and composition with a maze activity.

The activities presented in this paper, along with the observations reported by the undergraduate women involved, complement a long list of related projects and programs at other colleges and universities. Summer camps and programs designed to introduce K–12 girls to computer science concepts are becoming more widespread [15, 10, 4] and computer science faculty have a wealth of activities and presentations available to them that have been prepared, evaluated, and refined by their colleagues [2, 1, 22]. Further efforts to correct the gender imbalance involve comprehensive programs that link colleges and universities with their communities [21, 9, 5].

The structure of this document is as follows. Section 2 describes the workshop and activities in detail. Section 3 details the experiences of the undergraduate women involved in the outreach program and Section 4 details the experiences of the Girl Scouts involved in the workshop. Finally, Section 5 concludes the paper.

2. A COLLECTION OF ACTIVE LEARNING MODULES FOR K–12 STUDENTS

The following activities—designed wholly by undergraduate women—were designed to be components of a single workshop. However, each was designed so that it could be delivered in isolation.

Each of the activities designed for the workshops share one common objective—to present the girls with experiences that introduce to them the concept of "computational thinking" (*i.e.*, thinking abstractly and assertionally) [24] through active experimentation and feedback. Furthermore, each of the activities is unplugged both in the sense that computers are not necessary and in the sense that there is no syntactic barrier (in the form of a formal language) to the activities.

Additionally, each activity has multiple layers of difficulty so that they are truly applicable to K–12 students of all levels. In fact, the group of girls involved in the summer-camp workshop ranged from grades 4–8.

2.1 Guess Who?

In the classic children's board game Guess Who?$^{\mathrm{TM}}$each player is given a board with a set of characters on hinged holders (see Figure 1). Play begins by each player choosing a random character from a deck of cards. This character—the player's "secret character"—is not shown to their opponent. The game proceeds in rounds; in a round, a player asks her opponent a question about the opponent's secret character and eliminates possible characters on her game board based on the other player's response. The game is won when a player determines the other player's secret character.

This activity is a variant of the Guess Who?$^{\mathrm{TM}}$game. However, instead of students playing in rounds and selecting their next questions based on answers to previous questions, the students must select a set of yes/no questions *a priori* that are able to uniquely identify each possible character. In other words, the students must determine an enumeration from a set of yes/no answers to the set of characters.

Figure 1: The Guess Who?$^{\mathrm{TM}}$Board Game

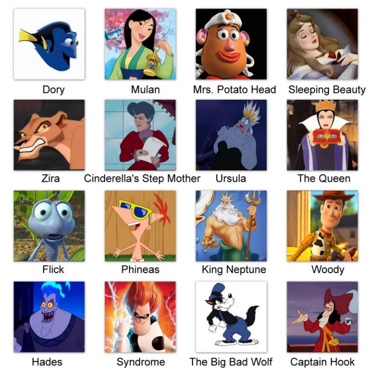

Figure 2: Guess Who Game with Characters Grouped by Characteristics

The objectives of this activity were to introduce the concepts of representation, information, and optimization; to practice assertional reasoning; and to discuss the concept of reduction.

Students were given game boards containing 16 characters that were chosen such that they could be represented by answers to 4 questions about "obvious" characteristics. In order to accommodate a range of ages, two variations of a game board were produced. In the first (Figure 2), characters are arranged by common characteristics, whereas in the second, characters are placed randomly.

Students were asked to fill out a sheet of questions that permitted them to ask at most 20 questions to their opponent. When students finished writing their questions, they were asked to pair up with another student and exchange their question sheets. A student was to think of a character, answer the question sheet honestly, and return it to her partner.

Students were then asked to revise their questions if they were unsuccessful in creating a set of questions that were able to identify their opponent's character. If their questions were successful, they were asked to try to reduce the amount of questions asked of their opponent. The students then played a second time; however, instead of thinking of a "random" character, they were asked to try to find a character that "broke" their opponent's questions.

All students were ultimately successful in creating questions that enumerated the characters. However, not all students were able to create an optimal number of questions.

This activity lends itself particularly well to lively discussion both during and after the activity. There are a great deal of questions that can be posed to the students, for example:

- Is it necessary to ask more than 16 questions?
- Are there four questions that will work?
- Can there be less than four?
- Does another set of characters exist that requires more than four?
- Does another set of characters exist that could be successfully enumerated with less than four questions?
- What is the relationship between four and sixteen?

In addition, this activity can be done in the context of facial recognition, which can lead a discussion about the similarity between facial recognition, fingerprint recognition, and DNA fingerprinting. In the presentation, this discussion featured the observation that Google Maps^{TM}mapping service and certain methods for DNA sequencing reduce to the shortest path problem.

2.2 Mazes and Robots

The barriers to learning how to program have been lessened by simple, accessible, visually-oriented programming languages like LOGO and Scratch. However, control structures like loops, conditional execution, and the idea of state are difficult concepts for students with no previous exposure.

The women chose to create activities that involved navigating a robot through a maze in order to introduce of the idea of programming as an activity involving the assembly of state-modifying instructions that, as a whole, map an input to an output. These tasks use a familiar model of state (the position of a robot in a maze) and familiar instructions (directional commands).

In the following activities, groups of students are given a floor-mat maze and a set of cards. Each card contains one of three commands: **go forward**, **turn right**, or **turn left**. Students then create a program (*i.e.*, a stack of cards) that, when followed by a student volunteer playing the role of a robot, navigates a maze.

Though the set of available instructions lacks selection or repetition, well-designed mazes can challenge students to use concepts like recursion and abstraction and to think about issues like resource utilization and efficiency. Indeed, these are some of the objectives of these activity.

2.2.1 *Activities for Young Students*

The act of solving a maze off-line and creating a set of instructions that can be followed to navigate the maze can be sufficiently challenging to provoke interesting points for discussion with young students. "Solving a maze" means a pencil-on-paper backtracking exploration to most young students, so the intellectual profile of building a precise set of instructions that guides directly from entrance to exit differs from their past experience.

Furthermore, opportunities arise to discuss other ideas in computer science. For example, mazes with multiple solutions can spur discussion about whether some are "better" than others (efficiency).

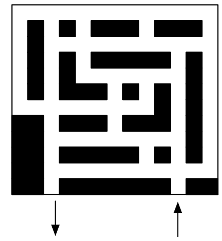

Figure 3: Robert Abbott's No-Left-Turn Maze

The following maze activities provide larger challenges and therefore expose other interesting, and sometimes deep, ideas to students.

2.2.2 *Multi-State Mazes*

Solving a traditional maze holds little challenge for older students. Multi-state mazes [20] on the other hand, use constraints on player movement to create mazes in which a player may be in the same position in the maze, but *not* the same state. For example, consider a maze in which the player cannot make left-hand turns or U-turns. When a player approaches an intersection in such a maze, the direction from which they approach the intersection determines the set of available directions that the player can travel. See, for example, Robert Abbott's no-left-turn maze [3] in Figure 3.

We used such a maze for this activity. A solution strategy that students can be encouraged to adopt is to recursively solve the maze. In other words, to guide the robot to the end of the maze from the start, first find a possible right hand turn that places the robot directly on the path to the exit. Once a suitable intersection is found, the problem has been reduced to guiding the robot to that location from the start. Students quickly adopted this strategy, even with minimal prompting. For example, prompting one group of 7th grade girls with "what does the *last* instruction have to be?" was enough prompting for the group to solve the problem recursively.

2.2.3 *Composition Mazes*

The specification for a set of instructions that solve a maze is obvious: if the player follows the instructions starting from the entrance, the instructions terminate with the player at the exit. In order to introduce the idea of specification and composition, small teams of students were given two maze specifications that, when their solutions were composed, solved a larger maze.

Students were given large mazes with entrances on the top and bottom of the maze and highlighted regions in the center of the maze. One team of students was then instructed to create a program that navigated their robot from the entrance to a highlighted region. Another team of students was to create a program that navigated the robot from a highlighted region to the exit.

To further illustrate the difference between a specification and a program, there were multiple highlighted regions in the center of the maze. The team starting the maze had a weak postcondition: rather than their program terminating with the robot in a specific region, their program simply required the robot to be in *any* of the regions.

The team ending the maze, therefore, had a correspondingly weak precondition: their program was required to guide the robot from any one of three starting positions to the exit of the maze.

Given that students had no instructions that permitted branching execution or loops, the maze contained isomorphic paths (in the sense that a single program could be used to navigate any of the paths) from the highlighted regions to the exit. The isomorphism of the paths was non-obvious, though, so students were forced to think abstractly about what it meant for instructions to solve a maze For example, instructions that correctly solve a maze might have a player make wrong turns and then backtrack, or an instruction—like going straight into a wall—might not affect the state of the robot.

This activity, like the others, contains layers of difficulty. Clearly, the team that must end the maze has a more difficult task than the team that starts the maze. Not only does this allow the activity to accommodate a range of ages, it is by design a source of discussion.

Students see firsthand that the more information they are given and the more freedom they are given, the easier a problem is to solve. Furthermore, if the starting team's postcondition is strengthened, their solution is no longer correct. However, if the ending team's precondition is strengthened, their solution remains correct. Similar to other activities [23], students see the practical application of Hoare's rule of consequence [17].

2.3 Discussion

These three activities form the basis of a 2.5 hour workshop. Though in a workshop, they are given in sequence and tied together through discussion and short talks, each of the activities can be presented separately. All contain elements that require students to think abstractly and assertionally, all illustrate fundamental ideas in the discipline, all contain several layers of difficulty, and all were received positively and enthusiastically when they were presented by the authors.

Section 4 contains the result of pre- and post-workshop surveys administered to a group of 46 girls at a workshop containing these three activities. Though none of the surveys assessed the learning of the girls at the workshop, they did assess the primary objective of the workshop itself: to undermine misconceptions of computer science as a viable field of study.

However, as the next section demonstrates, the students who attended the workshops were not the only beneficiaries of this program.

3. OUTREACH AS A TOOL FOR RETENTION

An outreach program controlled by undergraduate women benefits two groups: the girls that participate in the program's activities and the women themselves. The program founded by the authors grew over the course of months to include not just one undergraduate and faculty mentor, but four undergraduate computer science majors, who together make up the entire female population of the computer science program, as well as two mathematics majors.

Through informal interviews, the women reported three common reasons that participating in the program was a positive experience: a) they felt as though they were making a difference in the girls' lives, b) they learned about topics that they would not have otherwise encountered, and c) they felt more connected to the world outside of campus.

3.1 Changing Lives

> It's important that they interact with people they can relate to and know that they have a future in almost anything. I also enjoyed spending time with the students and being able to convince them computer science is fun and interesting, which in turn convinced me I was in the right field. *– junior computer science major*

> I feel like I am part of the solution to a problem that directly affects me. *– junior computer science major*

A common theme among the women was a sense of empowerment. They uniformly felt that they were making a positive impact on the girls involved by changing their perception of computer science. Furthermore, they reported that being in direct control of the program was an extremely important part of their experience; *they* were responsible for changing the girls' minds. To them, it would not have been as valuable of an experience had they merely been enlisted to help a faculty member.

They also felt that it was important that the girls were presented with non-stereotypical figures and, as young women, they were better able to relate to the girls. They were adamant from the very start of the planning process that they present the problem-solving aspects of the discipline—rather than the technical aspects—to the girls.

3.2 Going Deeper

> I feel more likely to continue in the field of computer science because I greatly enjoyed the activity, and I also learned more about computer science through helping the girls with the activity. *– freshman computer science major*

> To create this activity we had to take an in-depth look into the problem and learn as much as we could about it. While designing the activity, we also had to look at different applications of the problem in the real world. *– junior computer science major*

> It was a good experience for me and I hope in the future we can do more of these projects to teach girls how great computer science really is. *– freshman computer science major*

Running the outreach program also gave the undergraduate women involved the opportunity to research sub-disciplines that they would not have otherwise had the chance to study. This allowed them to study branches of computer science that were interesting to them, but may not be included in the curriculum. Beyond being introduced to new topics, they were also given an opportunity to participate in a research-like experience: they read papers, followed citations, looked for applications of ideas, and created novel work with sound theoretical underpinnings.

An activity the women designed for forty-five minute workshops, but is not included in the paper, involved girls writing strategies for an iterated Prisoners' Dilemma tournament that was presented as a game-show. One participant reported that it was an extremely valuable experience for her to look at the myriad applications of the problem and that it helped her understand the abstraction of the game-theoretic problem.

In order to write their talks, the women had to not only research specific problems and algorithms, but also had to think about computer science itself as a entity. The history of the discipline, the patterns of thought that are woven throughout the discipline, and the commonalities between sub-disciplines are sometimes masked by discrete courses and are sometimes opaque to novices. However, through their broad research, these ideas became parts of their presentations and incorporated into how the women viewed their studies.

3.3 Building Connections

> Participating in this activity gave me the opportunity to look into what computer scientists really do on a day to day basis in their jobs. I learned a lot about how computer scientists are addressing this problem and it made me feel more connected to the broader community of computer scientists. – *junior computer science major*

The women reported that being a part of the efforts to include more women in computer science also helped them to feel a deeper connection to the computer science community outside of the college campus. By reading technical and education research papers and by representing themselves and the college to outside organizations, they realized that there is much more to their field of study than what they are learning from their professors. This, in turn, invoked a sense of curiosity about the different opportunities that their computer science education would present to them.

Finally, the undergraduate women needed to understand precisely why they liked computer science in order to determine how to attract other women to the subject. It has been noted that women are generally attracted to computer science for different reasons than men [12]. Designing activities geared at getting other women involved in computer science forced them to reflect on what initially attracted them to computer science and how they could communicate how their studies are aligned with that attraction.

Finally, each of the women reported that participating in the program gave them a renewed interest in the subject.

4. RESULTS

The goal of the women's outreach program was to change the perception of computer science among K–12 girls and for the girls to consider computer science as a viable field of study. In order to measure the effectiveness of their workshops, the women administered a survey both before and after a 2.5 hour long workshop. The before survey featured two free-response questions and two Likert items:

1. What is the first thing that comes to mind when you hear "computer science?"
2. What do you think computer scientists do?

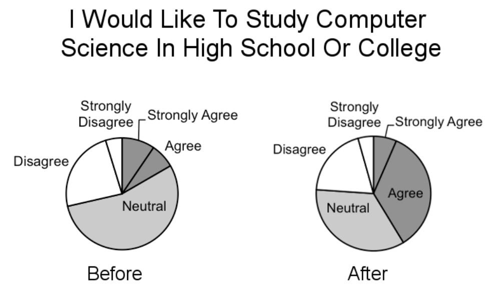

Figure 4: Responses to Question 3: "I would like to study computer science in high school or college."

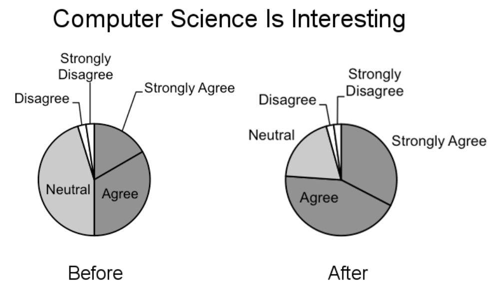

Figure 5: Responses to Question 4: "Computer science is interesting."

3. I would like to study computer science in high school or college. (Likert)
4. I think computer science is interesting. (Likert)

Question one received responses such as, "nerds," and "PowerPoints [sic], to make a website." Many responses to the second question were "study computers" and others such as "I think they learn about computers" or "I think they design microchips inside the computer."

The survey administered at the conclusion of the activity featured the same two Likert items along with a free response question: **"Has your opinion of computer science changed? If so, how?"** To this question, responses spanned a wide range:

- "I don't think it can change."
- "A little, I began to understand more."
- "Yes, I think that computer science is interesting, I might want to study it."

A vast majority of the answers were positive. Many were similar to the following: "I thought computer scientists sat at the computer all day." Another response was: "yes, because at first I thought that it's all about technology, but now I know it's much more."

The responses to the Likert items appear in Figures 4 and 5, respectively.

In the pre-workshop survey, "neutral" received an overwhelming majority of responses. Though the "disagree" and "strongly disagree" responses did not change dramatically in the post-workshop survey, the amount of neutral responses decreased sharply with a corresponding increase in "strongly agree" and "agree" categories.

The results of both the Likert item and free-response questions indicate that the workshop was successful in changing the perception of computer science for many of the participants.

5. CONCLUSION

Many K–12 outreach programs are designed and run by either computer science faculty or industry professionals with undergraduate students playing an assisting role. The authors have taken another tack. The program created by the authors has been successful in not only changing the opinions of K–12 girls about computer science, but also as an exciting opportunity for the young women involved.

Beyond the experience of the women, the activities presented in this paper can be presented without access to computers, either as a sequence of activities in a single long workshop or as the centerpiece activities of short presentations. The Guess Who? activity could even be used with slight modification in CS0 or CS1 classes. Materials needed to recreate the activities in this paper are available at `www.cs.moravian.edu/~lang/ugrad_outreach`.

6. REFERENCES

[1] Cs4fn: Computer science for fun. CS4FN Website. Retrieved November 15, 2010 from `http://www.cs4fn.org`.

[2] Women@scs computer science outreach roadshow. Retrieved November 15, 2010 from `http://women.cs.cmu.edu/What/Outreach/Roadshow/`.

[3] R. Abbott. Logic mazes. Website. Retrieved November 15, 2010 from `http://www.logicmazes.com`.

[4] J. C. Adams. Alice, middle schoolers & the imaginary worlds camps. In *Proceedings of the 38th SIGCSE technical symposium on Computer science education*, SIGCSE '07, pages 307–311, New York, NY, USA, 2007. ACM.

[5] A. Bruckman, M. Biggers, B. Ericson, T. McKlin, J. Dimond, B. DiSalvo, M. Hewner, L. Ni, and S. Yardi. "georgia computes!": improving the computing education pipeline. In *Proceedings of the 40th ACM technical symposium on Computer science education*, SIGCSE '09, pages 86–90, New York, NY, USA, 2009. ACM.

[6] T. Camp. The incredible shrinking pipeline. *Commun. ACM*, 40(10):103–110, 1997.

[7] S. Cheryan, V. C. Plaut, P. G. Davies, and C. M. Steele. Ambient belonging: How stereotypical cues impact gender participation in computer science. *Journal of Personality and Social Psychology*, 97(6):1045 – 1060, 2009.

[8] J. A. Cottam, S. S. Foley, and S. Menzel. Do roadshows work?: examining the effectiveness of just be. In *Proceedings of the 41st ACM technical symposium on Computer science education*, SIGCSE '10, pages 17–21, New York, NY, USA, 2010. ACM.

[9] M. Craig and D. Horton. Gr8 designs for gr8 girls: a middle-school program and its evaluation. In *Proceedings of the 40th ACM technical symposium on Computer science education*, SIGCSE '09, pages 221–225, New York, NY, USA, 2009. ACM.

[10] P. Doerschuk, J. Liu, and J. Mann. Pilot summer camps in computing for middle school girls: from organization through assessment. In *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, ITiCSE '07, pages 4–8, New York, NY, USA, 2007. ACM.

[11] M. Fellows, T. Bell, and I. Witten. Computer science unplugged, 2002. Retrieved November 15, 2010 from `http://www.csunplugged.org`.

[12] A. Fisher and J. Margolis. Unlocking the clubhouse: the carnegie mellon experience. *SIGCSE Bulletin*, 34(2):79–83, 2002.

[13] N. C. for Women & IT. Ncwit scorecard: A report on the status of women in information technology., 2007. Retrieved November 15, 2010 from `http://www.ncwit.org/pdf/2007_Scorecard_Web.pdf`.

[14] K. A. Frenkel. Women and computing. *Commun. ACM*, 33(11):34–46, 1990.

[15] D. P. Groth, H. H. Hu, B. Lauer, and H. Lee. Improving computer science diversity through summer camps. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, SIGCSE '08, pages 180–181, New York, NY, USA, 2008. ACM.

[16] B. Harvey. *Computer Science Logo Style: Symbolic Computing*. MIT Press, Cambridge, MA, USA, 1997.

[17] C. A. R. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10):576–580, 1969.

[18] K. Howell. The experience of women in undergraduate computer science: what does the research say? *SIGCSE Bulletin*, 25(2):1–8, 1993.

[19] D. J. Malan and H. H. Leitner. Scratch for budding computer scientists. In *Proceedings of the 38th SIGCSE technical symposium on Computer science education*, SIGCSE '07, pages 223–227, New York, NY, USA, 2007. ACM.

[20] E. Pegg. Math games: Multi-state mazes. MAA Website, 2003. Retrieved November 15, 2010 from `http://www.maa.org/editorial/mathgames/mathgames_11_24_03.html`.

[21] I. Pivkina, E. Pontelli, R. Jensen, and J. Haebe. Young women in computing: lessons learned from an educational & outreach program. In *Proceedings of the 40th ACM technical symposium on Computer science education*, SIGCSE '09, pages 509–513, New York, NY, USA, 2009. ACM.

[22] S. H. Rodger and E. L. Walker. Activities to attract high school girls to computer science. In *Proceedings of the Twenty-seventh SIGCSE Technical Symposium on Computer Science Education*, SIGCE '96, pages 373–377, 1996.

[23] P. A. G. Sivilotti and S. A. Laugel. Scratching the surface of advanced topics in software engineering: a workshop module for middle school students. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, SIGCSE '08, pages 291–295, New York, NY, USA, 2008. ACM.

[24] J. M. Wing. Computational thinking. *Commun. ACM*, 49(3):33–35, 2006.