Being an Undergraduate Teaching Assistant at Kalamazoo College: Collaboration and Professional Development

Torey Halsey

Advised by Dr. Pam Cutter
Department of Computer Science

A paper submitted in partial fulfillment of the requirements for the degree of Bachelor of Arts at Kalamazoo College.

2017-2018

Preface

I would like to thank all in the Computer Science department at Kalamazoo College for their efforts to outreach computer science education and experiential opportunities. I would like to acknowledge Dr. Pam Cutter, my SIP advisor, for her cooperation and advice during this entire process. I also would like to thank my family for all their support up to this point and beyond.

Contents

1 Introduction

1.1 The Mission Statement for the Computer Science Department at Kalamazoo College

I chose to attend Kalamazoo College (K College) to further my education and obtain a baccalaureate; moreover, the small college environment influenced my decision to apply to K College (K) rather than a large university. Upon coming to K, I was uncertain what curriculum I truly wanted to pursue. Although I had some background with database fundamentals (Oracle Database 11*g*) and some programming experience (Visual Basic 2012) through a one year database administration program at Wayne County Community College, I decided to explore other subjects during my first two quarters after I transferred to K. It was during the Spring term of 2016 that I registered for the only introductory computer science (CS) course offered that quarter. The course was instructed by Dr. Pam Cutter (co-chair of the CS department at K) and titled **COMP 107** *Pictures and Sounds: Programming with Multimedia with Lab*. The course introduced Jython; which is an implementation of the Python language seamlessly integrated with the Java platform. My reason for registering for a computer science course was to ultimately learn the object-oriented language Java; nonetheless, this introductory course taught me many valuable CS fundamentals. The fundamentals that a student can expect to gain from a CS course at K, even at the introductory level, are as follows:

> The mission of the computer science program at Kalamazoo College is to provide students with a foundation in the concepts and skills underlying computer science, in the context of an integrated, multi-dimensional liberal arts program. The program serves the needs of students at several levels, including students who take only one or two courses in the program, computer science minors, and computer science majors. At every level, the program is designed to expose students to the central theoretical concerns of this rapidly evolving discipline, and to generate an understanding of the principal techniques and algorithms necessary

> to support meaningful applications. The study of computer science enriches a liberal arts education by fostering skills in problem-solving, logical reasoning, organizing information to make it easier to understand and manipulate, expressing ideas precisely and effectively, and managing complexity.
>
> The computer science program is committed to providing a firm foundation in computing to our majors and minors, and to creating an environment in which students who do not think of themselves as computer scientists can learn about the discipline, discover an interest in it, explore, and thrive. Students who graduate with a major in computer science go on to careers in computing and to graduate school in computer science ("Program Information").

As the weeks progressed in COMP 107, I was enjoying the course to the extent that I considered pursuing computer science as a major. I was reaping the benefits of the mission of the CS department as I realized the importance of every assignment in my first introductory CS course at K.

1.2 Using Undergraduates as Teaching Assistants in Introductory Computer Science Courses

At many institutions, students interact with many members of a department staff rather than only the classroom instructor (Decker et al. 46). In addition to the department staff, CS students at K frequently interact with their peers in educational environments. Peer learning tasks can be used in a class, in a laboratory setting or as part of an out-of-class project (Wills et al. 6). Non-graded or only graded on participation tasks may consist of organized cooperative tasks that might end with a specific result that is shared with others (Wills et al. 6). Such cooperative tasks demonstrate the necessity to build upon effective communication skills and the ability to work well with others; moreover, these activities can change the dynamics of a traditional lecture class, prolong student attention, and increase a student's self-confidence to participate in discussions by answering as "we" rather than "I" (Wills et al. 9). When considering collaborative work,

we use “we” when formally representing a group to address ideas and not people in the process.

There have been various implementations of using students as teaching assistants (TAs)—such as using graduate students, undergraduate students, and even a hierarchy of computer science students (Vihavainen et al. 125). The use of teaching assistants in introductory computer science courses has been proven to be an effective inclusion for the learning benefit of all students (Vihavainen et al. 128). The benefit of being a TA is similar to the benefit of group composition: a TA is motivated to meet and work with students other than their friends while simultaneously benefiting the student and reducing the feeling of alienation for students without previous classmate connections (Wills et al. 18-20). A student with a TA position gains valuable experience on technical as well as inter- and intra-personal aspects of programming, such as communicating with people with different CS knowledge levels and experiencing the significance of best programming practices, all of which are important skills for a CS student (Vihavainen et al. 123).

1.3 Initial Experience with Teaching Assistants

COMP 107 introduced me to CS TAs. Although the TAs were not present on the first day of the course, Dr. Cutter (Pam) provided an “ice-breaker” activity for that day. All students in the course were encouraged to interact with one another through basic question and answer segments. Such activities allow students a chance to learn something interesting about their classmates and, through increased comfort levels, students tend to achieve more and develop working relationships and friendships across cultures (Wills et al. 15-16). Working relationships continued to blossom when the TAs

became involved in the mini-labs and weekly labs. As the course progressed and programming assignments were given as in-class tasks, the TAs would frequent the room and offer to provide their help. During the first week of the course, I rarely asked the TAs for their help to complete an assignment, but the generosity of the TAs offering their assistance inspired me to utilize the available help. When I needed assistance for further progress on an assignment, I was never given an answer by a TA, instead I was given suggestions such as what portion of my code may need checking and how to think through problems for a solution (as there were more than one solution available in most instances). Even when I was not receiving direct assistance from a TA, I was often helped indirectly by simply overhearing the conversations between TAs and my classmates. This was especially true during the TA evening help hours held on Sunday through Thursday from 7pm to 9pm. During these hours, two TAs were available to assist students with assignments from a variety of offered CS courses. Without the presence of a professor, TA evening help hours resembled peer-led team learning through cooperative learning without neither the requirement of an attendance commitment by students or a pre-planned agenda for the TAs (Roach and Villa 13.549.4-13.549.6). Such a relaxed learning environment provides a setting for students and TAs to learn from and develop alongside one another (Leyk et al. 1). This is especially true as some incoming students to the CS department may eventually become TAs.

Towards the end of the quarter, Pam suggested that I become a TA for the department and I accepted the position. Mentally preparing for what was to come, I reflected on the first day of the course—I learned the names of my classmates and we navigated the classroom as if we were all TAs. Moreover, my COMP 107 experience

with the upperclassmen TAs set an example of assisting students to reach a solution without simply giving the answer.

2 Being a Teaching Assistant

2.1 First Experience

My first CS TA position was for **COMP 105** *Introduction to Computer Science Using the Web* offered during the Fall quarter of 2016. This course also was instructed by Pam, who was the only CS professor at K that I had experience with so far. The topics of COMP 105 included the history of computers, what computers can and cannot do, the basic concepts of computer programming, program and user interface design, how computers represent information internally, an introduction to artificial intelligence, and the ethical and societal issues raised by the widespread use of computers.

While COMP 105 and COMP 107 are two of the three introductory courses offered by the CS department at K, they are respectively unique. I was familiar with some HTML5/CSS3 from COMP 107; however, the early focus on JavaScript in COMP 105 was a new experience for me. After learning the necessary JavaScript for the scope of the course, I soon realized that simply knowing the syntax of a language was not enough to be an effective TA.

My responsibility as a TA consisted of working two weekly labs. The first four labs required JavaScript, but the labs would usually build off of the mini-labs. Balancing my load of courses at the time and being an effective TA was a challenge my first time. It was during the lab of the second week that I experienced difficulty interacting with students because I did not fully prepare my knowledge of what questions I might encounter from students in the course. Although I familiarized myself with the syntax, I

constantly needed to refer to web resources when I assisted other students. I found myself spending more time helping one student than circulating the room and reaching more students. After the JavaScript labs ended, the remaining labs were topics that I had a better understanding of—such as Turing machines and constructing logic gates. However, I did not want to discourage students from continuing in CS because of lack of preparedness on my behalf.

Lack of preparation on my part quickly became a concern of mine; especially since I was fairly new to the curriculum. A study that examined the quantitative portion of student evaluations of TAs found a weak, negative correlation between the evaluations given for low-ranked TAs and the number of CS courses taken by students who received assistance from these TAs (Patitsas and Belleville 39). Being unable to assist some students forced me to ask other TAs or Pam for assistance in order to ultimately assist these students. In hindsight, such situations were great learning experiences for going forward in the curriculum. I learned that seeking the help of others was acceptable, even in my TA position; however, I needed to spend more time with the course material in order to develop fluency in the various topics in order not to confuse students or provide them with incorrect information.

2.2 Second Experience and Teaching Tips for CS TAs

Returning to K for the Winter 2016 term, the dynamic of the CS laboratory changed aesthetically and accommodated a lab series in **COMP 110** *Introduction to Programming with Lab*. The classroom incorporated a mural of an aquarium with various fish and sea décor. The motivation for such change was a result of a guest talk from Dr. Colleen Lewis of Harvey Mudd College. During the Fall 2016 quarter, Dr.

Lewis presented on changing learning spaces to be more inclusive for all students. Dr. Lewis explained how the labs of her college had murals on the walls, which impacted the atmosphere in the labs in a positive way. An aquarium theme was selected because COMP 110 is an introductory course, required of all CS majors and minors, in which students spend 4 weeks of the course on the *Aquarium Lab* series. The mural livened the lab and made it unlike the traditional lecture space. In addition, "Collaboration Center Hours" became a title replacement of the evening help hours held by TAs.

Attempting to create a culture and climate that supports all students and encourages diversity is what fueled creating the mural (Lewis 17). Our aquarium themed mural suggests a neutrality rather than a classroom that features portraits of historical faculty, as this may not always reflect ethnic or gender diversity. The feeling of belonging resonates as the mural is not randomly present, but it is applicable to an introductory theme for an incrementally building lab series. The mural can be thought of as an unofficial "ice-breaker" as students marvel at it upon entering the lab on the first day of their course.

This term I was a TA for COMP 105 again, instructed by Pam. Only one lab section was offered this quarter and I was responsible for assisting students along with two additional TAs. I was able to improve my knowledge of the topics; however, I had not made much progress on my teaching abilities. A new quarter meant a new set of students—some with prior experience, and some without any CS background—who were taking an introductory course in CS. When students would present questions about their erroneous code, I was eager to find the error for the student and attempt to explain what

occurred. I realized that this may create a dependency rather than teach the technique of actively engaging with the material first.

I learned that just because a concept had become clearer to me did not mean that I could simply restate it to a student seeking help. Instead, I needed to actively listen to the student in order to attempt to effectively assist them. The concept of *collaboration* made more sense when I was able to notice how helping one student could result in that student helping their neighbor and vice versa. Some students became as eager as to offer their assistance while I assisted a neighboring classmate.

In addition to giving a guest lecture, Dr. Lewis also provided a document on CS teaching tips for tutors. The three important actions for helping a student are introducing yourself, asking the student to describe their assignment problem, and asking the student to describe what they want help with ("Tips for tutors"). I find these three suggestions to have significant importance. Introducing yourself allows for the student to reciprocally introduce their self and this may aid the initial nervousness of asking a question, regardless of whether or not they have a question at that moment. Asking a student to describe their assignment problem presents an opportunity for them to verbalize what they are expected to accomplish. There have been instances when some students realize what step of the assignment they misinterpreted and then are able to continue without any formal assistance. Asking a student to describe the task that he or she wants help with insists that the student narrow in on what they are not fully comprehending. This also gives TAs a base to build follow-up questions for further assistance.

The document also provided next step guidance in the event that a student is unsure of where to even begin with their assignment. Rather than attempting to explain

what the assignment requires them to understand, it is suggested to ask the student to describe the problem in detail by asking them to explain the problem's goals, inputs, outputs, the algorithm that relates the input and output, and, if feasible, performing a handwritten example to ensure they have clarification ("Tips for tutors"). Some students need assistance with syntax errors or non-working code. Such situations provide opportunities for students and TAs to build upon their online help seeking tendencies by utilizing a search engine for suggestions on the Internet. Students often struggle to locate relevant information if improper queries are provided to the search engines; moreover, students may find some websites to be significantly helpful for future referencing (Hao et al. 21-22). Encouraging students to search online for help may influence a student's interest to perform an initial search while motivating them to spend more time on learning and often achieve superior learning outcomes (Hao et al. 25-27). Developing online help seeking proficiency is beneficial for all students in introductory CS courses, but will especially be of use to students who decide to continue into upper-level courses.

2.3 Third Experience

My third experience was more successful than my previous two experiences. I felt more confident in my ability to help students as a result of the experience gained from the previous two quarters; moreover, I had confidence in knowing that I was going to be a TA for COMP 107 with Pam's instruction—as I had performed well in the course the previous year. My goal as a TA during the Spring quarter of 2017 was to apply and build upon learned techniques for assisting students, connect with the course material at a deeper level, and constantly evaluate my interactions with peers (other TAs and students in the course).

My responsibilities were to assist with mini-labs twice a week and one of the weekly labs. As I assisted students, I found myself explaining the material even for the most trivial of questions. My intentions were to give clarification on the importance of a topic since a midterm and final exam are given in COMP 107 (there are no examinations in COMP 105). When I assisted students with their questions and concerns, I would listen closely to their responses and I would respond with positivity if they demonstrated that the solution to their problem was obvious or not as challenging as they perceived. I believe such interactions resulted as being more motivating than frustrating.

I learned the lesson of the importance of enhancing the reading culture while being a TA during this quarter. In COMP 107, there is a lab that requires students to re-type code obtained from a webpage into their integrated development environment (IDE). The specific situation I faced was with a student who was experiencing a syntax error that gave no specific information of where an error was occurring but that an error was present within the code. After reviewing the function that we perceived to contain the error, the student mentioned that he couldn't understand why the function did not work properly when he copied it directly from the PDF that was linked to the course web page. At this moment I realized the error and informed him that errors are likely to occur in this fashion when performing a copy and paste from a PDF to an IDE. While the error and the instruction were both subtle, providing a tip on the lab instructions that explicitly mentions the necessity to type the code into the IDE may have prevented the confusion. The purpose of typing example codes in the compiler instead of just copying and pasting them forces students to pay attention to the syntax of the programming language and implicitly asks students to be an active reader (Huan et al. 8). This situation

was an informal example of problem-based learning as we were able to actively engage with the content and effectively solve the problem and learn the reasoning behind why the directions stated to type the code (Fee and Holland-Minkley 129).

A challenge that I faced during this quarter was attending K's CS Collaboration Center Hours to seek help from TAs on assignments in my higher level CS course (**COMP 210** *Data Structures in Java*). Students in COMP 107 would attend Collaboration Center Hours and seek my assistance although I was not on duty. Rather than tell the student(s) that I was unable to assist them, I dedicated time to listen to their questions. Situations as such did not discourage me from attending Collaboration Center Hours but rather encouraged me to spend more time on the assignments that I was having difficulty with and utilize the office hours of my instructor. Meeting with Dr. Gerry Howser (the instructor for COMP 210) was a great experience because he demonstrated patience and was able to connect with my misunderstandings of the problems in the various programming projects in COMP 210. His method of asking questions to get an understanding of my comprehension of the topics resonated with me and provided a way for me to learn new techniques that I could use as a TA; moreover, this method proved the effectiveness of the aforementioned teaching tip by Dr. Lewis. In addition, his urging of online help searching helped me develop a better strategy for seeking help through websites such as StackOverflow and other computer science forum-based websites.

While simultaneously being a TA for COMP 107 and a student in COMP 210, I realized how the upper-level CS courses incorporated more problem-based learning. Rather than pursuing ideas with the specific prescribed approach that the instructor may have in mind, as in the introductory CS courses, students can pursue ideas

in a fashion that makes sense to them individually (Fee and Holland-Minkley 129). TAs were also not used as frequently in COMP 210 because *instructional scaffolding* begins to disappear once students complete the previous course in the curriculum (COMP 110). The instructors in the introductory courses support students cognitively, motivationally, and emotionally in learning while helping them to further develop autonomy (Vihavainen et al. 125). Instructional scaffolding is a process that dynamically changes and is dependent of the learners previous knowledge and progress (Vihavainen et al. 125). Specific to K, scaffolding subtly disappears in COMP 210 as the assignments near the end of the course provide less instructions and become more open-ended. As a result, this course allowed for collaboration between students for discussing what implementation methods were perceived as effective although the assignments required individual submission.

2.4 Fourth Experience

My fourth experience as a CS TA was with COMP 105 with Dr. Alyce Brady (co-chair of the CS department at K) during the Fall quarter of 2017. All TAs were encouraged to attend a meeting prior to the start of the quarter. This meeting utilized the CS Advanced Collaboratory—a room filled with CS books and miscellaneous CS information—that CS TAs are granted access to. Dr. Brady (Alyce) and Pam led the meeting with the assistance of Kristen Eldred, the Office Coordinator for the Computer Science, Math, and Physics Departments.

Although this was my third time as a TA for COMP 105, this was my first experience as a TA with Alyce (and my first experience as a TA without Pam). I developed a professional relationship with Alyce when she became my academic advisor

and I became a student in COMP 110. I also had experience with her teaching style from COMP 110. After having a successful quarter as a TA for COMP 107 and previous experiences with COMP 105, I expected to have just as much success as I did the previous term. Students from the previous COMP 105 offerings, Pam, my fellow TAs, and my own working on the mini-labs and labs taught me some common pitfalls to avoid. As a result, I became more effective at explaining programming concepts that needed to be mastered before later projects were assigned.

I did not experience many challenges this quarter because I was familiar and comfortable with the content of the course. Professional development was a goal of mine during this quarter. I felt more comfortable navigating the classroom and being accessible for students to ask questions. When I encountered students beyond the classroom, we would briefly talk about CS and how our CS courses were at that moment. This helped make connecting with students in the classroom more relaxing.

3 Future Teaching Assistant Opportunities

I plan to be a TA for COMP 110 during the Winter quarter of 2018 and also either COMP 107 or COMP 210 during the spring quarter of 2018. Having been a TA for four consecutive quarters has not only helped me to develop a better understanding of the fundamental topics in CS, but the TA position has taught me the importance of communication skills in regards to professional development. Knowing how challenging some of the introductory concepts are, even with some computer science background, has helped me to become a more patient explainer and better listener. Listening to a student explain—in his or her entirety—how they comprehended a topic helped me better assist the student. These interactions also presented an opportunity to reinforce a student's

knowledge of what they correctly understood, while teaching a topic that they may have not clearly grasped. Influencing students' success is a process that may involve multiple interactions, as noted by an educator:

> … you have explained something a thousand times but then when you do it again then you realise that, Ah, okay! I have to come up with a new way of saying this because this student doesn't get my other ways (Pears et al. 209).

4 Recommendations

Through my experience as a CS student and CS TA, the importance of actively reading the mini-lab and lab instructions are critical to having meaningful collaboration when questions arise. Urging that students always carry their textbook or assignment instructions and set priority for reading activities is a technique that is recommended for teaching computer science courses online in effort to enhance the reading culture; moreover, requesting that students type example codes into their IDE forces students to pay attention to syntax of the programming language (Huan et al. 8). Although K does not offer online courses, the same urgency to follow along with the textbook applies in all CS environments. From experience, this may help students prevent making non-syntactical errors and teach the recommended styles for writing code.

As the complexity of the assignments increases, using the Problem-Based Learning (PBL) pedagogy encourages students to engage with the content with personal experience while working through it with others, effectively solve problems and gain the corresponding knowledge, and PBL relaxes the traditional classroom environment so ideas can make sense to students individually. PBL centers student learning around open-ended, student-driven problems with the goal of students actively engaging with the information being taught. It is also noted that students will gain an appreciation for the

general skill of problem solving either later in the curriculum or upon entering the workforce (Fee and Holland-Minkley 131).

It may be worth providing a link to the mission statement within the course schedule page for each course offered, or at least the introductory courses to explicitly explain some of the skills a student should ultimately gain from any offered course in the department. The mission statement of the department can be viewed as just as important as the Honor Code of the college. Moreover, a physical copy of the mission statement could be provided and somehow incorporated into a syllabus quiz.

When considering TA training, having a "10 best/worst things" exercise can get everyone involved and add one thing (Reges 107). Future TA training sessions can revisit such a list and provide a way for educators and TAs to re-evaluate what progress is being made or new challenges that may have been encountered. TA training has the added benefit of familiarizing most, if not all, of the department to one another. In the classroom, this can create for exchanges between TAs and professors that can relax students and show them about a professor's priorities and lecture style in minutes rather than weeks of class (Dickson 77).

5 Conclusion

I have gained valuable experience as a CS TA at K; moreover, I hope that I have helped the students that I assisted to develop a maturity of the various concepts of the courses that I have worked with. TAs are able to leave such an impression on many of their students that they continue to seek advice and help even after introductory CS courses are completed (Decker et al. 49). Being able to convey computational ideas to others is certainly useful as a computing scientist, even if a teaching career is not pursued (Hug et al. 203). In the process of explaining new concepts to each new class, TAs are able to enhance their understanding of the concepts and improve their programming style (Roberts et al. 48). Potentially, effective use of TAs can improve the retention of women and underrepresented minorities in CS while reducing failure rates (Patitsas 115).

Through observing and being a TA, I have learned some important lessons. I have learned that being a good listener is essential for good communication. In order to assist a student to a solution, as a TA, it is necessary to thoroughly understand what difficulty the student is encountering; even if this requires asking multiple questions. The use of undergraduate students as TAs provides benefits over the use of graduate students because of the similarities in schedules, familiarity with the material and teaching techniques of the faculty, and also the potential to build a community environment between upperclassmen and underclassmen; especially at small colleges (Dickson et al. 168). There are a variety of CS knowledge backgrounds in CS introductory courses at K. TAs must be able to talk to the students in the course regardless of background differences and this helps to improve the TA's communication skills and build their confidence (Dickson 79).

6 References

Decker, Adrienne, et al. “Through the looking glass: reflections on using undergraduate teaching assistants in CS1.” *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*, 2006, pp. 46-50. doi:10.1145/1121341.1121358.

Dickson, Paul E. “Using undergraduate teaching assistants in a small college environment.” *SIGCSE ’11 Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, 2011, pp. 75-80. doi:10.1145/1953163.1953187.

Dickson, Paul E., et al. “Using Undergraduate Teaching Assistants in Small Classes.” *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, SIGCSE ’17, 2017, pp. 165-170. doi:10.1145/3017680.3017725.

Fee, Samuel B., and Holland-Minkley, Amanda M. “Teaching computer science through problems, not solutions.” *Computer Science Education*, vol. 20, no. 2, 2010, pp. 129-144. doi:10.1080/08993408.2010.486271.

Hao, Qiang, et al. “Predicting computer science students’ online help-seeking tendencies.” *Knowledge Management & E-Learning*, vol. 9, no. 1, 2017, pp. 19-32. https://www.researchgate.net/publication/315043681_Predicting_Computer_Science_Students'_Online_Help-Seeking_Tendencies.

Huan, Xiaoli, et al. “Teaching computer science courses in distance learning.” *Journal Of Instructional Pedagogies*, vol. 6, 2011, pp. 1-14. https://eric.ed.gov/?id=EJ1097042.

Hug, Sarah, et al. “Learning to love computer science: Peer leaders gain teaching skill, communicative ability and content knowledge in the CS classroom.” *SIGCSE ’11 Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, 2011, pp. 201-206. doi:10.1145/1953163.1953225.

Lewis, Colleen M. “ACM RETENTION COMMITTEE: Twelve Tips for Creating a Culture That Supports All Students in Computing.” *ACM Inroads*, vol. 8, no. 4, 2017, pp. 17-20. doi: 10.1145/3148524

Leyk, Teresa, et al. “Structured Peer Learning Program - An Innovative Approach to Computer Science Education.” *Texas A&M University Case Study*, 2017, pp. 1-6. https://www.researchgate.net/publication/314948271_Structured_Peer_Learning_Program_-_An_Innovative_Approach_to_Computer_Science_Education.

Patitsas, Elizabeth. “A case study of the development of CS teaching assistants and their experiences with team teaching.” *Koli Calling ’13 Proceedings of the Thirteenth Koli Calling International Conference on Computing Education Research*, 2013, pp. 115-124. doi:10.1145/2526968.2526981.

Patitsas, Elizabeth, and Belleville, Patrice. “What can we learn from quantitative teaching assistant evaluations?.” *Proceedings of the Seventeenth Western Canadian Conference on Computing Education*, WCCCE 2012, 2012, pp. 36-40. doi:10.1145/2247569.2247582.

Pears, Arnold, et al. “What's the problem?: Teachers’ experience of student learning successes and failures. *Proceedings of Seventh Baltic Sea Conference on Computing Education Research*, Koli Calling ’07, vol. 88, 2007, pp. 207-211. http://dl.acm.org/citation.cfm?id=2449323.2449353.

"Program Information." *Computer Science*, Kalamazoo College, 2017, http://www.kzoo.edu/programs/computer-science/.

Reges, Stuart. "Using undergraduates as teaching assistants at a state university." *ACM SIGCSE Bulletin*, vol. 35, no. 1, 2003, pp. 103-107. doi:10.1145/792548.611943.

Roach, Steve, and Villa, Elsa. "Enhancing peer led team learning through cooperative learning." *Paper presented at 2008 Annual Conference & Exposition, Pittsburgh, Pennsylvania*, 2008, pp.13.549.1-13.549.10. https://peer.asee.org/3707.

Roberts, Eric, et al. "Using undergraduates as teaching assistants in introductory programming courses: An update on the Stanford experience." *ACM SIGCSE Bulletin*, vol. 27, no. 1, 1995, pp. 48-52. doi:10.1145/199691.199716.

"Tips for tutors." *Tips for tutors*, National Science Foundation (NSF); Sagefox Consulting Group; Harvey Mudd College. http://csteachingtips.org/tips-for-tutors. Accessed 29 December 2017.

Vihavainen, Arto et al. "Massive Increase in Eager TAs: Experiences from Extreme Apprenticeship-based CS1." *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '13, 2013, pp. 123-128. doi:10.1145/2462476.2462508.

Wills, Craig E., et al. "Studying the Use of Peer Learning in the Introductory Computer Science Curriculum." *Computer Science Education*, vol. 9, no. 2, 1999, pp. 71-88. doi:10.1076/csed.9.2.71.3811.