HIGH SCHOOL COMPUTER SCIENCE EDUCATION

A Master's Thesis Presented to the Faculty of

the Department of Computing Sciences

Villanova University

In Partial Fulfillment

of the Requirements for the Degree of

Master of Science

in

Computer Science

by

Samantha Bewley

December 2018

Under the Direction of

Mary-Angela Papalaskari

Table of Contents

Acknowledgements

I would like to thank my thesis advisor, Dr. Mary- Angela Papalaskari. She has provided unending guidance, suggestions and the answers to all my questions, making this work possible. Her advice and encouragement has been invaluable. I would also like to thank Dr. Spyros Doukakis for his guidance with creating the questionnaire to interview the teachers, and his feedback throughout the project, including providing qualitative research guidance. I would also like to thank Dr. Robert E. Beck for his help with the IRB process. I would also like to thank Dr. Jeffery Popyack for his support and his help in facilitating the interviews of the teachers at the Drexel ReThink program. I would like to thank Villanova University for providing the funding for my Master's Degree in Computer Science. I would like to thank all of my teachers, at all stages in my education for their encouragement.

I would also like to thank the High School Principals that allowed me to interview the teachers and provided me the contact information for the Computer Science teachers. I would like to thank the teachers for agreeing to be interviewed, without them and their invaluable opinions and explanations, I would not be able to complete this thesis.

I would like to thank my place of employment, eMoney Advisor, and my manager, Mark Stoughton for providing me the flexibility needed to complete this thesis and

continue working full time. I would also like to thank my coworkers for their support.

I would like to thank my friends for their support and encouragement, not only during this project, but in all my endeavors. I would like to thank my father, Joel Bewley for helping me edit my thesis. I would like to thank my mother, Devika Gill, my father, Joel Bewley, and my three younger sisters, Olivia Bewley, Alexandra Bewley, and Sophia Bewley for their love and support, throughout this project, and my entire life.

Finally, I would like to thank my fiancé and future husband, Kyle Abramowitz for his never-ending love, not just during this project, but always. Words cannot express how thankful I am for all of his support, especially during some of the more stressful times during this project.

Abstract

One of the challenges in the field of computer science is teaching the subject at the high school level. Thirteen computer science teachers, one technology teacher and one department chair for technology were interviewed to determine how they thought computer science education could be improved at the high school level. The qualitative research addressed curriculum, professional development, educational computer science standards and frameworks, technology, and pedagogy. Institutional Review Board approval was obtained for the research. Nvivo was used to analyze the interviews. When the results were compiled, many teachers were concerned that there were low numbers of students interested in computer science. Having low numbers or students enrolled in computer science classes contribute to low numbers of computer science teachers. Different way to address these problems are proposed.

Introduction

The field of Computer Science is constantly changing and growing in order to solve new and exciting problems using technology. One of the challenges in the field of computer science is teaching the subject, especially at the high school level. Since high school computer science instructors are the ones in the classroom teaching, their perspective is valuable when looking at computer science education at the high school level. The teachers know what is effective and what is not, and also have suggestions about how the topic should be taught and what curriculum should be utilized.

To gain these insights, high schools in Montgomery, Delaware, and Chester Counties in Pennsylvania were emailed to ask their high school computer science teachers to participate in this qualitative study. Additionally, teachers at Drexel's ReThink summer program were asked to participate in this study. A total of thirteen computer science teachers, one technology teacher and one department chair for technology were interviewed and their responses were compiled to determine how they thought computer science education could be improved at the high school level.

There were three main issues that the teachers brought up multiple times. The first was the lack of other computer science teachers which meant teachers did not have as many peers with whom they could collaborate and brainstorm. Another concern was that the number of students expressing interest in the field was low. Lastly, many computer science teachers became the instructor for their school because no

other faculty members showed interest, or none were qualified. A strong effort needs to be made to encourage more high school students to enroll in computer science education in order for the schools to justify providing additional staff members and resources toward the programs.

Methodology

Each teacher was asked questions from a questionnaire that was used to guide the conversation. The questionnaire was designed to address curriculum, professional development, educational computer science standards and frameworks, technology, and pedagogy. The questionnaire that was used to conduct the interview is attached in Appendix A. Each interview was guided by the responses given by the teachers, so not all questions on the questionnaire were asked, and not all interviews touched on concepts in the same order. The goal of the interviews was to cover each topic in the questionnaire.

Institutional Review Board (IRB) approval was obtained for the study in order to protect the rights of human subjects involved in the research activities. An expedited review was applied for because the research involved interviewing subjects. Since the interviews took place at the teachers' schools, first the principal of the school needed to be contacted to obtain permission to contact the computer science teacher and the contact information of the computer science teacher. Wikipedia was used to obtain the names of the high schools in Delaware, Chester and Montgomery counties and the emails for the principals were found using Google. A sample email that was sent to the principals is included in Appendix B and a sample email sent to the teachers is attached in Appendix C.

After the interviews were completed, the interviews were transcribed and then analyzed using Nvivo, which is a qualitative data analysis software. There is a

transcript of two interviews included in Appendix D. Nvivo was used to categorize what was said in the interviews, those categories were then used to draw conclusions from the interviews. Since the interviews are anonymous; each teacher is referred to by a letter. The interviews were conducted in August and September of 2018.

A convenience sample was used to contact the teachers. Since Villanova University is in Delaware county, teachers in Delaware county and two neighboring counties, Montgomery county and Chester county were interviewed. The reasoning behind this was due to needing to travel to the sites to interview teachers. In addition, this allows teachers in the same area to be studied, in order to see how teachers in Southeastern Pennsylvania feel about computer science education. Also, different states have different requirements and guidelines regarding computer science education at the high school level, so the research was limited to one state. Using a convenience sample does introduce a bias into the results, as it is likely that teachers in the same geographical area would have some similar opinions.

High School Computer Science Standards and Frameworks

Background

There are no universal standards or frameworks for teaching computer science. The Computer Science Teachers Association (CSTA) has developed standards that are starting to be accepted and used by many different states. The CSTA supports and promotes the teaching of computer science for K-12 teachers, in addition to providing opportunities for teachers to better prepare themselves to be successful instructors. The organization presently has more than 25,000 members from over 145 countries, some of whom are high school teachers (ABOUT CSTA). The CSTA standards introduce the fundamental concepts of computer science to all students, promote computer science classes at the secondary level to fulfill a computer science, math or science graduation credit, encourage schools to offer additional secondary-level computer science courses and increase the availability of rigorous computer science instruction for all students, especially those from underrepresented groups. If the CSTA Curriculum Task Force approves content, it can be claimed in alignment with the CSTA K-12 Standards. The standards were written by educators and are to be used by teachers, administrators and policy makers (CSTA K-12 Computer Science Standards). There are five main concepts for the standards - computing systems, networks and the Internet, data and analysis, algorithms and programming, and impacts of computing. There are different grade-level goals for each of the concepts (Progression Chart - CSTA K-12 Computer Science Standards). The CSTA standards are a core set of learning objectives for a complete computer science education (CSTA K-12 Computer Science Standards).

Since CSTA promotes the teaching of computer science, the goal was to understand how high school teachers felt about the standards and whether they utilized any of the standards in their curriculum. Input from teachers is critical in order to understand how the standards are interpreted and used. The CSTA standards are detailed and measurable which allows a teacher to determine how students are preforming. The standards also provide the teacher with more direction in order to develop curriculum.

The K-12 Computer Science framework was developed by the Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and the National Math and Science Initiative. States, districts and the computer science education community were also involved in creating the framework. The goal of the K-12 Computer Science framework is to develop conceptual guidelines for computer science education. "The framework is designed to guide computer science from a subject for the fortunate few to an opportunity for all" (K–12 Computer Science Framework). The framework is more of a high level-guide while the CSTA standards spell out more specific goals. The main goal of the K-12 computer science framework is for students to engage in the concepts and practices of computer science in order to develop a foundation of computer science knowledge. Additionally, it is expected that this knowledge will lead to students being able to solve problems in new ways, as well as become creators of computing technology through the use of computational thinking. Since the K-12 framework is

a high-level guide, it is meant to be used by states, districts and organizations implementing computer science information. It's an organizing structure on which to base standards. There are also parts of the framework for curriculum and professional development. The K-12 framework is centered on research and practice in computer science education, including work from CSTA, International Society for Technology in Education (ITSE), the Advanced Placement (AP) Computer Science Principles curriculum framework, the ACM for curriculum guidelines for undergraduate computer science programs, and frameworks from other countries (Hendrickson, A Vision for K–12 Computer Science). This is important because the K-12 framework is not trying to replace other frameworks, but to build on the previous research and effort already being put into computer science education.

Since there are so many frameworks and standards devoted to computer science education, the K-12 computer science framework tries to tie them all together with a high-level overview. Depending on the background of the teacher, he or she might prefer a more high-level overview depending on his or her level of familiarity with computer science and his or her ability to develop curricula. This may also allow teachers to be more flexible in choosing the most effective teaching methods.

Results

When teachers were asked about different standards, teachers had mixed feelings. Teacher A thought the standards were confusing and hard to find and therefore did not use standards in preparing her curriculum. Teachers A, D, E, F, H, and K also

recognized that there was not a single, defined set of standards to use. Teacher M, thought that since there was no unified curriculum for computer science, then the standards must have been written so broadly that they were not very meaningful. He said, "The only time I've used them is when you have to give something to a principal, so you put something on top." Teacher N developed her curriculum and then found out that the CSTA standards reaffirmed her curriculum.

Teachers A and E did not feel the standards were useful; they felt that way because there were multiple standards, so when they tried to use standards or look for standards to use, there were many options. However, teachers M and N, who did not like the standards did use them when they had to present something to a principal or the administration. This shows that the standards are confirming what teachers are currently teaching if they are able to use them to justify their curriculum. Teacher N said she "went through and looked very closely at those and it was reaffirming that everything I was already doing, I can pull their standards out and this is exactly what we're doing." Teachers H, K, L, and N already had a curriculum that followed the standards, but it can be concluded that these standards could help schools that presently have nothing in place.

Teachers D and F, who teach AP Computer Science A and AP Computer Science Principles used the College Board standards for their teaching. Another teacher who was a part of CSTA, teacher H, talked to other teachers in an attempt to get feedback from his peers to make sure he was relaying high-quality information to his

students. He also used the K-12 Computer Science framework to develop additional classes for his department, which has students from pre-K through grade 12. He thinks the framework is also useful for teachers who are developing new classes for different levels in their districts; it provides guidelines that are aimed at different grade levels. Teacher L said that being a member of CSTA Philly is also a good way for high school teachers to connect with peers from other schools to brainstorm and discuss changes in curriculum and content.

Another member of CSTA, teacher K, prefers the CSTA standards, not only because she is a member, but also because she knows how the standards were written and developed. They were written collaboratively and sent out to the entire membership months in advance for further revisions and clarification, and Pennsylvania also adopted the standards. Many additional states have also adopted the CSTA standards. Another member of CSTA, teacher L, likes the CSTA standards but also appreciates what ITSE is doing. Since CSTA standards are so lengthy ITSE is trying to promote a smaller one-page document standard. She believes that schools should meet the ISTE standards and then look at the CSTA standards and that ITSE can be used as a stepping stone.

Discussion

At the beginning of 2018, Pennsylvania endorsed the CSTA K-12 standards. The Pennsylvania Department of Education plans to provide schools with guidance on implementing the standards with a toolkit and information guide (Gov. Wolf

Applauds Board of Education for Computer Science Standards). Naturally, members of CSTA endorse the standards put out by CSTA, so it makes sense that teachers who are involved with CSTA like the standards. However, as mentioned in the results section, other teachers interviewed did use the standards when they needed to either reaffirm their curriculum or when presenting something to administrators. Even though teachers may not use the standards from the beginning, the fact that they can map their curriculum to the standards means the standards are useful and informative. Also, Pennsylvania and many other states have currently adopted the standards, which demonstrates that they are being used as intended and accurately communicate computer science concepts.

College-Level Computer Science Classes in High Schools

There are two main programs that offer college level computer science classes at the high school level, International Baccalaureate (IB) and Advanced Placement (AP) classes.

International Baccalaureate

The IB program encourages students to think independently and drive their own learning, become culturally aware by learning a second language, and be able to engage with people in a globalized, rapidly changing world. The IB program also leads students to some of the highest-ranking universities around the world (Benefits of IB for Students). There are over 4,500 schools worldwide that are IB World Schools. The IB Diploma Programme (DP) is for high school students aged 16-19. The DP has six subject groups; Computer Science is part of the fourth group, which is the experimental science group (Diploma Programme, Computer Science in DP). There is a standard-level (SL) and higher-level (HL) computer science course offered through the DP program. There is a focus on computational thinking, supplemented by programing. This allows students to learn programming skills. Developing programming experience will allow students to develop higher-level thinking skills. Also, teachers will be able to select the programming language they feel is best for the course. Algorithmic thinking is only assessed using pseudo code. This means that for the algorithm portions of the assessment the syntax of the

programming language is not tested, but the student's ability to think algorithmically is tested (Computer Science in DP). The difference between the SL and HL courses is that the HL course has three additional topics with a recommended 45 additional teaching hours. The HL/SL core consists of four topics that take a recommended 80 hours to complete. The four topics are system fundamentals, computer organization, networks, and computational thinking, problem solving and programming. The HL extension topics are abstract data structures, resource management, and control. There is also a case study, which is different each year, with an additional recommended teaching time of 30 hours. An optional SL/HL core exists with a recommended teaching time of 30 hours, and an HL extension with 15 hours recommended teaching time. The HL extension has four options for the student to choose from - databases, modeling and simulation, web science, and object-oriented programming. There are 30 recommended teaching hours allocated to the internal assessment solution, which is "Practical application of skills through the development of a product and associated documentation." The final recommended 10 teaching hours are for the Group 4 projects. At the end of the course, there are three tests that the student must complete along with submitting written commentary and the Group 4 project (International Baccalaureate Diploma Programme Subject Brief).

In order for a school to be an IB World School, the teachers must attend professional development workshops provided by the IB. Teachers must attend an IB Category 1 workshop in their subject. The IB does not provide the curriculum, but rather there

are standards in place that must be met. IB teachers also have access to the IB online curriculum center where teaching materials are available. There are also online forums for the program topics where IB teachers participate (International Baccalaureate Diploma Programme Subject Brief).

Since IBCS is not offered in as many Southeastern Pennsylvania high schools as AP Computer Science, it is important to understand the key differences and how teachers feel about each set of standards and tests. The IBCS standards are overarching and the curriculum must ultimately be approved by the IB organization. This provides the teachers with the ability to create their own curriculum based on the general topics. It makes sense that teachers might like to have the freedom to create their own curriculum, and not have to teach from a strict curriculum. Since no curriculum exists, it is important to understand how the teachers view the standards, and whether they deem them to be adequate.

As stated above, teachers who wish to teach IBCS must attend a professional development workshop before starting. However, the IB recommends that teachers continue professional development for as long as they remain involved with computer science. Professional development is an essential part of teaching, and it is important to understand if and how teachers continue their professional development after the required IB Category 1 workshop.

Advanced Placement

Advanced Placement (AP) courses are high school courses that allow students to earn college credit. This allows students to save money on college tuition, study abroad or obtain a second major (Work Toward College Success). AP classes also allow students to skip introductory classes and start taking mid-level classes (Work Toward College Success). AP classes allow students to build skills necessary for college, such as time management and study skills (Work Toward College Success). Similarly to IB, the AP program is a global program. There is an Advanced Placement International Diploma (APID) that is available to students attending secondary schools outside the United States, and students studying in the United States and applying to schools outside of the country (AP Around the World). AP classes end with a college-level assessment that is then graded by college and university faculty, along with experienced AP teachers. This assessment allows students to demonstrate their knowledge of the subject. Over 90 percent of four-year colleges and universities in the United States award credit, placement or both (AP Computer Science A Course Description). If a school wants to offer AP courses, they must participate in the AP Course Audit. College faculty members review the class syllabi in order to provide teachers and administrators with clear guidelines on curriculum and resource requirements for AP classes. This allows colleges and universities to confirm that those classes meet or exceed the college and university expectations (AP Computer Science A Course Description).

There are two AP computer science classes offered, AP Computer Science Principles and AP Computer Science A. AP Computer Science A is about the fundamentals of programming and problem solving. AP Computer Science Principles is about the fundamentals of computing, including problem solving, working with data, understanding the Internet, Cybersecurity and programming. AP Computer Science A has one exam at the end of the year, which is multiple choice and free response. AP Computer Science Principles has two projects during the course and one exam at the end of the year, which is multiple choice.

AP Computer Science A is an introductory computer science course taught in Java. The goals of the course are for students "to be able to design, implement and analyze solutions to problems, use and implement commonly used algorithms, use standard data structures, develop and select appropriate algorithms and data structures to solve new problems, write solutions fluently in an object-orientated paradigm, write, run, test and debug solutions in the Java programming language, utilizing standard Java library classes and interfaces from the AP Java subset, read and understand programs consisting of several classes and interacting objects, read and understand a description of the design and evolution process leading to such a program, and understand the ethical and social implications of computer use." (AP Computer Science A Course Description). The AP Computer Science A course also must include at least 20 hours of structured, hands-on lab experiences. There are three example labs on the AP Computer Science website that have been developed for teachers to use. Teachers are able to use the provided labs, develop their own

labs, or use labs from textbooks and other sources. There are also sample AP tests online for teachers to use in their classes and to help prepare students for the test (AP Computer Science A Course Description).

AP Computer Science Principles is supposed to be equivalent to a first-semester introductory college computing course. The AP Computer Science Principles curriculum framework is comprised of many parts. Computational thinking practices, seven big ideas, learning objectives, essential knowledge statements and exclusion statements comprise the curriculum framework. The computational thinking practices are connecting computing, creating computational artifacts, abstracting, analyzing problems and artifacts, communicating and collaborating. The seven big ideas are creativity, abstraction, data and information, algorithms, programming, the Internet, and global impact. These are ideas that are fundamental to studying computer science. These big ideas contain enduring understandings, which are core concepts that the student should know. The enduring understandings have one or more learning objectives, which are things students are expected to be able to do by the end of the course. The learning objectives contain essential knowledge statements, which are facts that students should know which show that the students understand the learning objective corresponding to the essential knowledge statement. The exclusion statements provide clarity about the scope of a learning objective or essential knowledge statement (AP Computer Science Principles Course and Exam Description). At the end of the course, there is a multiple-choice exam on the course learning objectives. In the two course

performance tasks, students explore the impacts of computing and creating computational artifacts digitally and through programming. The course does not require a specific programming language; the instructor is able to choose the programming language or languages that will be used. The instructor is also free to choose the instructional approach. As with AP Computer Science A, instructors are provided with sample AP exam questions and performance tasks (AP Computer Science Principles Course and Exam Description).

There is also professional development offered by the College Board for AP teachers. There are workshops that the teachers are able to attend, and online training (Professional Development | AP Central – The College Board). Teachers are also able to enroll in AP mentoring which pairs teachers who enroll with expert AP teachers to provide support. Mentees chose an AP mentor, who has a group of up to four people who are being assisted. "Research shows that peer mentoring can be highly effective in supporting a teacher's practice when it's sustained over the course of the school year and provides opportunities for reflection and application." (Enroll in AP Mentoring | AP Central - The College Board). AP mentoring can enhance teaching skills for AP teachers by focusing on the requirements. Teachers who participate get personalized feedback to be applied in the classroom, support from other teachers, and also attend an AP summer institute or an AP one-day workshop (Enroll in AP Mentoring | AP Central - The College Board). Workshops aim to help teachers prepare students for success, develop instructional approaches that align with AP goals, identify areas where students may need more preparation, and draft a

syllabus for the course. AP Summer Institutes provide teachers with at least 30 hours of training in order to strengthen their ability to teach AP courses. There are also curriculum and professional development opportunities made available by other organizations that are endorsed by the College Board (About AP Workshops | AP Central - The College Board).

There are several different AP classes offered, and the goal of the question was to understand whether teachers are teaching AP Computer Science classes and gauge their like or dislike for the classes. There is professional development that is offered, but not required by the College Board, so the objective was to see if high school AP teachers are taking advantage of the professional development opportunities. If a school offers both AP Computer Science A and AP computer Science Principles, it would be helpful to determine what the teacher thinks about the differences between the two courses and whether they are beneficial to the students. One would think that the same teacher would teach both classes, but that might not be the case, assuming that the school offers both courses.

Background

Since there is an exam at the end of both the IB and AP programs to gauge student abilities, one of the goals of the questionnaire is to understand whether teachers felt like they were teaching to an exam and not properly instructing their students to understand the material. Teaching to the exam, or item teaching is when teachers base their instruction around test items and test questions. Curriculum teaching is

when teachers instruct to a body of content or a set of cognitive skills that will be on the test. By item teaching, a student's knowledge of the subject will not be adequately tested, and the teachers can feel pressure for their students to do well on the test (Popham). It is important to understand the teachers' views on whether they believe they are teaching to the exam at the end of the course, or if they are teaching the content of the class.

Results

Teachers were asked if students need to take AP Computer Science A and AP Computer Science Principles in a certain order. All ten schools (A, B, C, D, E, F, H, K, L, M) that offered both AP Computer Science A and AP Computer Science Principles allowed the students to take both classes in any order. Five teachers (B, C, D, H, K) suggested that students could take AP Computer Science Principles first and then AP Computer Science A, but it depends on the students' backgrounds with computer science and whether the classes fit in their schedule. Teacher K said that "A is a difficult class, so I recommend it to a lot of my intro students to take principles before A when I didn't think they were ready for A." Two of the teachers (J, O) taught at schools that did not offer AP Computer Science classes.

Many of the teachers appreciate AP Computer Science A but have mixed feelings about AP Computer Science Principles. Teacher A views AP Computer Science Principles as "more of an intro to computer science that even a non-computer science major might take." Some teachers (F, N) didn't feel that AP Computer

Science Principles correlated to a college class. According to teacher L, AP Computer Science Principles is more of an introduction to the big ideas of computer science and is more accessible to more people and provides a great foundation if the student wants to continue in computer science and take AP Computer Science A. Teacher M stated "I'd like it if it actually gets students interested in computer science. I don't like it if it's presented as a real alternative to computer science." Teacher M also said AP Computer Science Principles seemed more like "computer science light." Teacher N believes AP Computer Science Principles is comparable to a middle school class and is a watered-down version of a computer science class without much value for students who "have no intentions of going into engineering or computer science." She also felt that the computer science principles class is not painting a complete picture of a computer science class.

Ten of the teachers who taught AP Computer Science A (B, C, D, E, F, H, K, L, N) thought that it was a good class and believed that the content and the curriculum were things that students needed to know in order to advance in the computer science field. Teacher L who teaches AP Computer Science A at a high school and was an adjunct at a college feels that the AP Computer Science A class is the same class as an introductory college computer science course.

Eight teachers (A, B, D, E, K, L, M, N) have also attended professional development provided by the College Board or have graded the exam for the College Board. Teacher B likes grading the AP exam because it is more collaborative than an online

professional development class. Grading the AP exams is another form of professional development that teachers enjoyed. Teacher D stated that grading is "a form of professional development where you can interact with other teachers. Computer science teachers are often the only computer science in their school and it's a way for us to get networking." Grading the AP exam also gives teachers the opportunity to see how other students, besides the students they teach are answering the exam questions. As three teachers (B, D, E) pointed out, it also allows them to collaborate and interact with their peers and other computer science teachers, which is necessary and something they are not necessarily getting in their day-to-day routine as a high school teacher. Teacher K explained that a weeklong summer institute for AP Computer Science A "was overwhelming. It was way too much information for one week for a brand new computer science teacher."

When asked about teaching to the test, eight (A, C, D, E, F, H, K, L) of the teachers involved with the AP Computer Science are mindful that there is a test at the end of the year, but there are very mixed feelings about the test. Teacher A stated that she is teaching the content, but the content is on the test, and therefore she is mindful that she is getting them ready for the test. Similarly, teacher D makes sure that she covers all the concepts, since most students are planning to take the test and the goal is for the students to do well. She also feels that the AP exam at the end of the year is very similar to the tests she would give throughout the year and uses AP style questions on her tests. Teacher K teacher believes that the things that are being taught in the course are items that students need to know in order to advance

in the field of computer science. However, she doesn't believe that students should be handwriting code, but it does help to make sure that the students really know what they are doing and not relying on an integrated development environment (IDE). Teacher L thinks that the test is appropriate, so teaching to the test is not a problem. However, there are things that she feels the need to teach that are on the test that are not practical. Conversely, there are things that she thinks students need to know that are not tested which she teaches. Teacher E emphasizes the concepts that are on the test, following the curriculum of the College Board. However, she also relays tips that are specific to the test in order to try to help the students do better on the test.

Teacher N refuses to teach to the test, so she concentrates on computer science fundamentals and believes that if her students can master topics and in-depth concepts, and believe they are successful programmers, then they will pass the exam. She doesn't want to take the approach of having students pass the exam and not become good programmers. Curriculum teaching is when teachers don't feel like they are teaching to the test but believe the concepts that on the test are important to teach the students.

Because AP Computer Science Principles is a new class, teacher G feels like the subject matter is too expansive to teach to the test. However, she spends summers grading the test, which allows her to know exactly what the College Board is looking for. But the feedback shows that other teachers feel like they have to teach to the

test. Teacher H stated that there is so much material that needs to be covered that he is forced to teach to the test and doesn't have time to let students thoroughly explore the concepts. Instead he believes they need to move on to the next topic in order for students to be adequately prepared for the test. Teacher B "spent like a solid month testing free response question, multiple choice question, free response question, multiple choice question, and it just got into a boring pattern". In order to prepare his students for the test, he felt obligated to teach to the test. However, the same teacher does like the AP classes.

Teacher J who doesn't teach AP Computer Science or IB classes believes teaching to the test is not a good way to learn and doesn't ever want to teach to a standardized test of any kind. In his opinion, under no circumstances can teaching to an exam be considered an effective manner for students to learn new concepts.

Discussion

None of the teachers who were interviewed taught IBCS, so no conclusions were drawn that related to that instruction. However, ten of the teachers who taught AP Computer Science A (A, B, C, D, E, F, H, K, L, N) liked the format of AP Computer Science A, so it is reasonable to believe that adding an additional college level course that would create extra work and require more professional development is not a practical idea, especially when many teachers are satisfied with AP Computer Science. Teachers already have a heavy course load and, since all teachers interviewed were teaching AP Computer Science or wanted their schools to begin

teaching AP Computer Science, adding an additional course for college credit would create unnecessary stress.

The teachers who liked AP Computer Science Principles did so because it was an introduction to the field that students of all backgrounds could take, and the overarching principles of computer science were taught. However, the complaint from teachers who did not like the class was that it was a watered-down introduction that didn't offer much value.

The goal of AP Computer Science A is for the class to be equivalent to an introductory college-level computer science class, so the fact that a teacher who has experience with both considered them to essentially be the same class means that the college board is meeting this goal.

Many of the teachers that were interviewed liked the format of the AP Computer Science test. The teachers who feel as if they are teaching to the exam seem to understand the challenges of the AP tests. The teachers feel the need to constantly keep moving and teaching the next concept in order to get the students ready for the test, and they are frustrated by this dynamic. It is beneficial that the students will be ready to take the test in May, however teachers are unable to go more in-depth on topics that students may be interested in or struggling with. Even so, this problem exists with many curricula and classes. High school and college-level teachers and professors must be prepared to cover certain content before the class is completed.

The AP summer institutes provide 30 hours of training and instruction for a week, which can be very overwhelming for a teacher who has never taught AP Computer Science before. Having so much information can be counterproductive because the teacher may become overwhelmed and not retain as much of the information.

Professional Development

Background

Professional development is needed to help teachers learn and refine instructional strategies needed to teach the complex skills students need to learn in order to succeed (Darling-Hammond). Effective professional development is structured professional learning that improves student-learning outcomes and advances teacher practices (Darling-Hammond). In order to be effective, professional development needs to be content focused, incorporate active learning, support collaboration, use models and modeling of effective practices, provide coaching and support, offer opportunities for feedback and reflection, and be of sustained duration (Darling-Hammond). The professional development that is provided by the different course options and frameworks is valuable to high school computer science teachers because it can help them become more confident in their teaching and also improve their skills. It will be important to understand what teachers think of the current state of professional development and also what teachers believe is lacking in that area.

The K-12 computer science framework also includes guidelines for professional development. These include customizing professional development in order to meet teachers' various backgrounds (Hendrickson, Teacher Development). Not all computer science teachers have the same education and background. Some may never have attended professional development courses for computer science. It is important for professional development instructors to understand the different

backgrounds of the participants in order to provide the most effective instruction. This would include addressing anxiety that may be expressed by novice teachers who may not be very knowledgeable about computer science (Hendrickson, Teacher Development). Professional development instructors should strive to understand the teachers' present level of knowledge and build upward from there. The professional development provider should connect professional development experiences to a curricular context (Hendrickson, Teacher Development). This way, teachers can easily use what they learn from the professional development session in their classroom. Professional development should include a focus on increasing access and equity (Hendrickson, Teacher Development). The last standard is that professional development should address the management of a productive computer lab environment (Hendrickson, Teacher Development). Computer science education is very hands-on, and teachers should be able to manage a computer science classroom in order to make sure the students participate in the most effective manner possible.

Results

The teachers interviewed have different backgrounds. Seven teachers have bachelor's degrees in science or math subjects (A, B, E, F, H, J, M). Additionally, teacher K has a bachelor's degree in business with a marketing concentration, and a bachelor's degree in business education. Teacher L has a bachelor's degree in MIS, and teacher M has a bachelor's degree in math with a computer science concentration since computer science wasn't available as a degree. Teacher C and

teacher N have a bachelor's degree in electrical engineering. Teachers also learned computer science in various ways. Nine teachers (A, B, C, D, F, H, K, L, N) went into industry and became familiar with computer science there. Teacher D graduated with a bachelor's degree in secondary math education and joined a company as a programmer (there was no computer science degree at the time at her college). She worked her way up as a software engineer, then completed a master's degree in math education and started teaching math before gradually adding computer science classes. Since there is not enough demand for computer science classes, she is still also teaching math. Of the teachers who had master's degrees, two (A, F) of those were in math education, two (E, H) had taken classes towards a master's degree in computer science, one (N) had completed a master's degree in computer science, three (D, L, M) had a master's degree in education, and teacher C has a master's degree in multicultural education, teacher K has a MBA.

There are a lot of different professional development opportunities available for computer science teachers. As discussed in a previous section, the K-12 framework also provides guidelines for professional development. It is important to understand to what extent the teachers believe that they were assisted by professional development. It is also valuable to explore what they believe is missing from professional development and whether that can be changed. There are many different organizations that provide professional development for high school teachers. Almost all teachers (A, B, C, D, E, G, H, J, K, L, M, N, O) who were interviewed had undertaken some form of professional development. Three

teachers (C, D, H) think one of the best things to come out their professional development was interacting with other teachers. As mentioned before, most of the time a school only has one computer science teacher, maybe two, so interacting with other teachers is important during professional development workshops. Teacher C liked professional development because he liked interacting with other high school computer science teachers. However, he also felt that "each teacher is pretty much isolated in their school and there isn't really much of that strong community in this area."

Teachers also had ideas on how to improve computer science professional development. According to teacher E, all the professional development is driven by outside forces. He would like to see in-house professional development for math or science teachers to learn how to incorporate computer science into their curriculum. Teacher K commented that there is not one place to see what different professional development is available. However, she stated that CSTA is "hoping to get a website up and the information in from lots of different sources with CSTA Philly to really list what's out there."

Three teachers (H, L, M) also attended conferences. Teacher H went to a conference that included both high school and college professors and thought it was beneficial to sit down and talk with both people from each group. Teacher H also said he thinks it would be beneficial if professional development sessions had time to get

into groups and apply concepts that were just taught in a workshop, instead of rushing to the next workshop.

Discussion

Teachers who presently teach computer science come from different backgrounds and may not necessarily have degrees in computer science. The teachers interviewed attended many different professional developments, completed additional schooling and undertook self-teaching that was done in order to for them to teach computer science. Not having a traditional background in computer science should not be a barrier to being a high school computer science teacher. However, the lack of standardization does create the worry that there will be to a high variance in the quality of teaching the subject, especially if under-qualified teachers are asked to teach computer science.

The teachers who were interviewed seemed to have no problem finding professional development opportunities. One of the key features of computer science professional development is the ability of the teachers to interact with their peers. This is something that is missing in many of the teacher's day-to-day routines. This is different than with subjects, such as mathematics, where there are several teachers covering the same subject, and even the same classes. The research suggests that this is due to the lack of computer science teachers. Since there are not that many teachers, then it stands to reason that the registrants for training events

would be comprised of a small group of people who regularly see each other at these events.

Some teachers also mentioned that they would like more time during professional development events to sit with their peers and talk about applying concepts and ideas that were just discussed. This is important because many teachers are not able to sit down and brainstorm with another computer science teacher at their school. This can also help teachers to absorb more of the information and discuss how they want to incorporate the new topics into their curriculum.

Technological Pedagogical and Content Knowledge Framework

Background

The Technological Pedagogical and Content Knowledge (TPACK) framework measures a teacher's knowledge with respect to technology, pedagogy, content knowledge, and the combination of these areas (Doukakis et al.). Content knowledge (CK) is the knowledge that teachers have regarding Computer Science. Pedagogical knowledge (PK) is the knowledge of educational process and the methodology of teaching and learning. Technological knowledge (TK) is a teacher's ability to manipulate technological tools and utilize technology in the classroom (Giannakos et al.). PCK is a framework, which includes the knowledge of the content and the knowledge of pedagogy. Teachers who have good PCK are able to transform their knowledge of the content and make it accessible to the students (Doukakis et al.). Instructors need to have a grasp of the content knowledge, and the goals, objectives, means and strategies of teaching programming, and the knowledge of how to evaluate scientific literature on programing and teaching approaches (Doukakis et al.). Since computer science is a changing field, it is important that teachers feel as though they are prepared to teach the content of computer science. There are also several different standards, some mentioned here, and teachers should be able to navigate and use them to help determine their curriculum. However, it appears that this may be overwhelming or difficult for a teacher depending on his or her background or educational level. A teacher may have the best intentions of teaching computer science but may struggle with the content and the best way to relate that information to students. The goals of the content questions are to understand how

teachers feel about their level of knowledge in the field of computer science and their opinions on the multiple standards. It is also important to determine whether teachers believe there are any topics or other things that are missing from computer science education.

The focus of educating teachers has shifted to focus on pedagogy (Mishra and Koehler). Approaches of teacher education have emphasized either the knowledge domain or the knowledge of pedagogy (Mishra and Koehler). The goal of PCK is to blend both together (Doukakis et al.). It seems that most of the high school teachers will believe that they do not need support in pedagogy, but in blending pedagogy with the content, in this case computer science.

Results

Teachers were asked to describe their teaching style and how they assess their students in order to get a better understanding of their method of teaching computer science. There are many different ways that teachers assess their students. All teachers give their students tests and quizzes, with a midterm and final. Two AP computer science teachers (E, H) make their tests and quizzes similar to the AP exam in order to prepare students. Five (E, F, G, H, K) teachers also used rubrics to grade projects and labs. Two (E, M) teachers had students hand write code on their assessments to make sure they were keeping up with the concepts. Eleven teachers (A, B, D, E, F, H, J, K, M, N, O) of the teachers believed that projects and labs were an integral part of computer science teaching.

Thirteen teachers (A, B, C, E, F, G, H, J, K, L, M, N, O) emphasized the importance of interacting with students and not just lecturing, those same teachers were also very hands-on. Teacher B stressed the importance of students collaborating, that "students in the room learn as much from their peers as they could possibly learn from me. And then the other thing is I don't want to be the one who limits what they can learn." He wants to be a resource for the students. He also wants to make sure the students can actually do the work and not just copy down the homework from the Internet. Teacher L uses guided and open inquiry in her classes. The students come up with their own questions and she helps them locate the resources necessary to formulate the answers.

Those thirteen teachers also said it was important for students to code and write code and learn from their own mistakes. All of the teachers interviewed also had some sort of project-based classes, where they would introduce a concept and then have students solve a problem and complete a lab or project related to the concept. Teacher J stated it is important to help students get fired up and ignite their passion.

Teachers were also asked about curriculum, not what they teach but how they develop their curriculum. All but two of the teachers being interviewed teach computer science, teacher O teaches a technology class, and teacher I is the department chair for technology. AP Computer Science teachers have topics that they must cover in order to prepare students for the AP exam.

Twelve of the teachers (A, B, C, D, F, G, H, J, K, M, N, O) preferred to be able to develop their own curriculum with access to resources than be given a curriculum that they must follow; two teachers (E, L) had no preference. Teacher L who had seven courses in her rotation focused on developing and updating one course each year. When she first started teaching, she felt that she had to write all of her own curricula from scratch and couldn't use anything that was pre-made. Teacher C is "very happy making my own curriculum at this point. When I first started, having something to follow is very important." Teacher D used an approved pre-made curriculum for the Computer Science Principles class and she "found it really helpful to have those lessons that outline some examples of some assessments to use."

Teacher N was hired the week before classes started and wasn't given any curriculum. She developed the classes by working backwards from where she wanted them to be when it was time to take the AP exam. Her students take two classes before they take AP Computer Science. She wrote the entire curriculum for three different classes in a year, day by day. She liked being able to develop her own curriculum because it gave her the freedom to teach the students what they needed to know for both the AP exam and beyond. She came from industry and also just completed a master's degree in computer science, so she felt able to teach the students what they needed to know.

Teacher J liked to develop his own curriculum and enjoyed the freedom to change it every year to improve the class. Teacher L liked that when she developed her own curriculum, she could tailor it to the interests of the students. Teacher H liked to develop his own curriculum because he was able to incorporate problems that related to the real world and keep up with current trends in the field. He was able to do this while meeting the AP Computer Science requirements. Teacher K worked with a university to offer dual enrollment with the computer science class. This was achieved by adding in topics that was required for the university to the AP Computer Science class in the fourth marking period after the AP exam. Since she developed her own curriculum, she was able to easily incorporate the extra topics.

Teachers were also asked about different technologies used in their classrooms. Many students have their own computers and the teachers used many different technologies, smartboards and different ideas. Teacher H had a 3D printer and a Glowforge that he incorporated into his classes. He also used Arduinos to incorporate robotics and wearable technology into his teaching. There were many different technologies mentioned by the teachers that they use. All of the teachers were excited to integrate different technologies and excite the students about computer science.

Teachers were asked what they do if they do not know the answers to the questions posed by their students, and how they get up to speed on new topics in the field of computer science. Teacher A teacher sometimes Googles it right then and there if

the student comes up with something she is not familiar with, or she will either look at it or have the student do some research and they will discuss it the next day. She also keeps up to date with current events and tries to incorporate current events into their curriculum. Teacher F teacher looks up tutorials on YouTube and researches the question, usually on the spot. Teacher B who teaches an advanced seminar also relies on YouTube lets students decide the topic. He then utilizes various online resources and tutorials to get up to speed on the subject matter. He incorporates YouTube, Udemy and Udacity into his instruction. He is also looking at MIT and Stanford to search for new ideas to teach his students. He believes learning together is a great approach to encourage students to participate in the discussion. Online resources are very good for teachers to utilize when they are unsure of a question or topic being posed by a student. Online resources are also an effective approach for teachers to remain current on emerging computer science topics and trends.

Teacher E teacher informs her students that she is not an expert and will sometimes be learning along with them. Teacher H is not afraid to jump right into questions posed by students and researches the idea to make sure that what the student is proposing is feasible. It is sometimes effective, but he sometimes must bring up the question during the next professional development opportunity to obtain the answer needed to assist his students. Teacher J will go and teach himself what the student wants to know and then teaches the student; he lets his professional development go in the direction that the students are interested in. If a student asks

about something he is unfamiliar with, he takes that and uses it to enhance his skill set and then shares that knowledge with the class.

Teacher M acknowledges that some students have greater knowledge than he does about the field. If they ask him a question, he likes to sit down and talk through it and figure it out with the student. He also likes to research it online. He believes this is also good because in the real world there is not one person who knows every aspect of computer science, and students will often have to find the answers on their own. He likes to learn and add a new thing every year.

Teacher G makes sure she is constantly learning and participating in professional development. She also tried to focus on one topic or class every summer. She had a student years ago who wanted to take AP Computer Science A. However, the school didn't offer the class at this time, so she helped the student find resources and let her use her classroom to work toward that goal. Even though the teacher didn't have the background necessary to help, she was able to provide resources for the student. Teacher D used to work in the industry and still has a passion for computer science believes teachers must be creative in their efforts to support students who have an interest in computer science.

Teacher C goes to workshops at different colleges. Teacher L attends professional development seminars every summer, and various conferences and workshops every year. Teacher N went back to college and earned a master's degree, but still

plans to take courses in areas where she is not familiar. Now that she has completed her master's degree, she no longer qualifies for tuition reimbursement and must pay for additional classes on her own. She also plans to take free or low-cost Massive Online Open Courses to keep herself current in the field. She took a cybersecurity course as an elective during her master's degree undertaking since her students expressed interest in that area.

Discussion

Allowing teachers to develop their own curriculum, or to do so with additional resources, allows them the flexibility to cater their curriculum to their students' interests and also spend greater time on areas where students might be struggling and need more time to grasp the concepts. Some teachers also liked that there was some flexibility after the AP Computer Science exam in May to go over additional topics that were not on the exam. This allowed for greater discussion on areas in which the students were interested or were necessary to provide dual enrollment credit to the students. For a teacher who is familiar with the topic, being able to develop an original curriculum provides the freedom to teach the content they believe will be most useful. While writing their own curriculum is good for teachers who want to have that flexibility, teachers should also be aware that there are good resources available to them and shouldn't feel pressured to write their own curriculum from scratch, unless they want to.

The teachers who were interviewed were confident in their pedagogical knowledge. They had a process for teaching computer science and felt that it worked well for them. Almost all of the teachers stressed that relying only on lecturing was not a good way for them to teach computer science. Instead, they believed hands-on learning, lab work and project work were a better way to teach and for students to really grasp the concepts. Since they are teaching high school classes, all teachers also gave tests and quizzes to test the students' mastery of the subject. Since computer science is so hands on, it is important for students to learn by performing and by writing code. This allows students to think critically about problems and the best ways to solve them.

Teachers also needed to keep their content knowledge and skills up-to-date and persistently stay abreast of the latest concepts in a field that constantly evolves. This enables teachers to obtain the necessary knowledge to answer the questions posed by their students. Even so, there may be questions posed by students that may give teachers guidance in the direction they need to take to ensure that they have the latest training necessary to effectively guide their students. It can be concluded that it is important for students to understand that the teachers will not know the answer to every question. In the field of computer science there will always be questions and new things to learn and this gives students a realistic view of the field. It is also good to help students learn how to find the answers to their questions and not rely solely on the teacher for the answers.

Future Goals of Teachers and Current Issues

Results

When asked about the future of computer science education, the teachers had varied answers. Eight teachers (A, B, C, D, E, G, H, I) wanted more students to take computer science classes. Teachers A, and J advocated for coding to be taught in the required technology course. Teacher A stated, "there is really no coding in it." Teacher E called for students to be introduced to computer science in the early grades, as well as for more women and underrepresented groups to be encouraged to embrace the field. Teacher J said that education needs to recruit more people who are qualified to teach computer science but understands that competing with the salaries offered in the industry presents a major challenge. This sentiment was echoed by teacher J who said that in addition to the allure of much higher compensation, a career in the industry allows those trained in computer science to work as coders, programmers and developers, and that those with skills in these areas might be hesitant to switch careers to become teachers.

Teacher C would also like to see more buy-in from the school administration. He thinks the students seem to really enjoy computer science once they get into it, and most of the administration is supportive of computer science. However, he would like to see more support, such as requiring every student to be enrolled in a computer science course if they have the required math prerequisites. His proposal would require students to opt out of the computer science class if they are not interested in taking it, instead of opting into the class. He also thinks it's important

for students to be exposed to computer science before college, especially for students who are planning to choose it as a major.

Teacher M who teaches at a school that offers three years of computer science, including one course after AP Computer Science, wants to see a fourth year of computer science added. Five teachers (B, D, E, G, I) at schools that offered various numbers of computer science classes would also like to see more courses offered. Teacher D commented that she "would like to see more students taking them [computer science classes]. I'd like to see more courses offered not just two AP classes in my school." This would require the schools to put more resources into the computer science programs.

Teacher B said it would be helpful for students to have some experience in computer science prior to entering the workforce. Every field is touched upon by computer science. His program reached only 10% of the students at the school. He would like to see enough students express interest that it would justify the hiring of another teacher. He also wants to integrate computer science into other classes, such as astronomy, or environmental science.

Teachers were also asked about what was currently missing from computer science, and how they view the future of the field as it pertains to high school education. Teacher A cited a lack of buy-in from administration. She also wanted teachers who were more prepared to teach computer science. Teacher B said more qualified

teachers are needed, and while trainings and workshops are necessary, so is the need to recruit teachers with a background in the field. However, he doesn't want a hard limit to the requirements because that could exclude qualified candidates. He thinks there is no good answer to this problem. Teacher K teacher thinks some sort of certification would be good as well as more defined criteria for becoming qualified to teach computer science.

Teacher N said there are not enough computer science teachers, leading schools to resist offering the classes. Teacher J teacher commented "a fundamental problem with computer science education is the people who are most qualified to teach it aren't because they're making more money out in industry as coders, as programmers, and developers." Yet another teacher, teacher L, said she believes the lack of teachers is one reason why schools are hesitant to embrace a robust program, or even to offer classes at all. She would like to see it become integrated in the curriculum beginning at the elementary level.

Teacher D said students only take it if they have an interest because it is not a requirement. Teacher C teacher wants to attract a greater number of female students. He also believes that students with strong math backgrounds could be easily persuaded to give computer science a try.

Teacher M said there was not enough buy-in from the administration, and many administrators don't understand the field, and very few teachers have a formal

computer science background. He would like to see more people involved who actually understand computer science and are excited about the possibilities of passing the knowledge along to students. Teacher A said the administration "recognizes the need for computer science and for STEM, but actually getting it implemented and finding the time to plan that has been difficult." Teacher J, who didn't have a background in computer science wanted to also make sure that he keeps growing in the field.

Teacher H believes the subject needs to be more approachable for students and not so overwhelming. The idea is to get them to become intrigued and inspired so they do not become intimidated. Teacher H wanted to focus on emerging technologies, such as AR, VR and Bitcoin and Blockchain. It would appear that many students might not take computer science if they are intimidated by the topic, and this may contribute to low enrollment numbers in some high schools. If students can become interested and curious about computer science, it will not seem as intimidating and they might discover that they enjoy the field. Introducing new and exciting technologies can be another way to get students interested in the field of computer science.

Discussion

One of the main problems addressed by the respondents was the lack of computer science teachers. It can be concluded from this information that this is also closely

related to the lack of students embracing computer science classes at the high school level. One of the reasons schools only have one or sometimes two computer science teachers is because there are simply not enough students taking the computer science classes to need more teachers. Some teachers who were the only computer science teacher at their school didn't even have enough students interested in the field to justify them being designated as a full-time computer science instructor. They were required to teach classes in other subjects. A second problem with needing more teachers is that many of the people who are qualified to teach computer science chose instead to go into industry and become software engineers instead of becoming teachers.

Computer science teachers risk feeling isolated among their peers because they often have no colleagues who teach the same subject. This has been found to be the case with those who teach art. They may be predisposed to more feelings of isolation than other subject-area teachers due to the specialized nature of their subject area, the lack of a supportive network within the school building, and the scheduling of their classes (Sindberg). Even though this is about art teachers, the same thing can be said about computer science teachers. There are many parallels between the two subject areas in how they are approached in education. Both art and computer science teachers are often the only teachers in those subjects at their school and teach elective classes, not core subject classes such as math and science. Because of the specialization of computer science, many teachers are not able to talk about teaching strategies and topics within computer science because others don't

really understand the topic. There are also problems with students finding time in their schedule to take the classes. This dynamic feeds on itself, and since there are a limited number of students taking computer science, there aren't as many classes being offered. Computer science is often considered an elective, and therefore viewed as less important than math or science.

There are several different ways to go about enrolling more students in computer science classes. In the state of Pennsylvania, there are currently no requirements for any computer science classes needed to graduate high school (Pennsylvania Code § 57.31). It is reasonable to believe that adding a requirement for students to take even one year of computer science would mean that every student is exposed to computer science. It also may inspire students to continue to take more computer science classes and spark an interest in the field. Another option at the school level and not the state level is for administrators to be more involved in computer science education. The interviews with the teachers have shown that many teachers are more than qualified to teach computer science. However, many times the administration just lets them do their own thing and doesn't endorse or support computer science as much as some other subjects.

Another option is to find ways to encourage students to take a computer science class to determine if they will be interested in computer science. Teacher C commented, "I think if they tried it, I think they'd like it." One idea is requiring students to take a class unless they opt out, this was proposed by Teacher C to his

administrators, but it was not implemented. It can be concluded that Teacher C's proposal could be a very effective way of exposing more students to computer science. Underrepresented groups, minorities and women also need to make up more of the computer science classes, which is a completely separate concern.

More students need to get involved in computer science. Studies have shown that high school girls have identified that they are not interested because of a lack of female role models, limited or no knowledge of the application of computing, interest in things other than computers and negative perceptions of computer scientist as nerdy (Heersink and Moskal). These feelings and conditions also apply to male high school students. The stereotypes around computer science as a field need to be changed in order for students to become more interested. Even though high school students have grown up with technology and video games and therefore view technology as fast and exciting, their first programming experiences are tedious and boring. Many students also lack confidence in their basic programming skills (Heersink and Moskal). By incorporating new and exciting concepts to replace those that are not as exciting, students might be more inclined to embrace computer science.

While many teachers focused on the future of computer science and getting more students and teachers involved, it is also important for teachers to make sure that they continue to improve their level of knowledge about the field and the most effective methods to teach students.

These current problems reflect what teachers are seeing today and have shaped their answers in regard to what they would like to see in the future of high school computer science education. It makes sense that encouraging more students to take computer science classes will create the need for more teachers. Students who are part of underrepresented groups, minorities and girls should also be encouraged to take computer science, but care should be taken to not exclude males or students who are not minorities nor identify with an underrepresented group. However, if more students are taking computer science, the problem of not having enough qualified teachers to teach will be worsened.

Another solution could be a state regulation requiring that students learn computer science. Legislation has been passed to count computer science coursework toward a math or science graduation requirement in public and charter high schools in Pennsylvania (Orenstein). However, there is still no high school computer science graduation requirement in Pennsylvania.

In the meantime, those interested in high school computer science education must persist in their efforts to have their district administrations support the field. As part of that, administration officials need to understand that since computer science is constantly changing, teachers must have access to professional development in order to become and remain effective instructors.

Conclusion

There are currently no computer science requirements to graduate high school in the state of Pennsylvania. However, the state of Pennsylvania has endorsed the CSTA K-12 standards.

Thirteen computer science teachers, one technology teacher and one department chair for technology in three counties in Southeastern Pennsylvania were interviewed to understand their current feelings and opinions on computer science education at the high school level. A qualitative study was done and teachers were asked questions about their curriculums, professional development, educational computer science standards and frameworks, technology and pedagogy. The interviews were analyzed in order to see how teachers want to improve computer science education at the high school level.

The three main topics that many of the teachers noted were the feeling of isolation since there are not many computer science teachers. Teachers are also concerned that the number of students expressing interest in computer science is low. Lastly, many of the teachers feel there is a lack of qualified computer science teachers.

Computer science, and more specifically computer science education, is constantly changing. One of the main changes that teachers would like to see is for more students to take computer science classes and become excited by what the field has to offer. This in turn would create the need for more teachers. There have been

many standards, curriculums, different classes and new ideas proposed and implemented, but without more students taking the classes, and thus requiring more qualified teachers to be hired, these efforts will never reach their full potential. Another component of any future effort would be putting thought into how to better market the field of computer science to encourage more students at the high school level to become involved.

In forming this conclusion, it should be noted that a voluntary response bias exists because the results only include teachers who chose to participate. In the future, a shorter interview could be conducted with a more accurate representation of the high school teachers, and also over a larger area instead of only three counties in Southeastern Pennsylvania.

Appendix A – Questionnaire

- Tell me about your computer science course.
- How did you decide the curriculum and what to teach?
- What are the main topics in your curriculum?
- Technology
    - Which technologies do you use in your classroom?
    - How do you use technology in your classroom to teach computer science?
- Pedagogy
    - What is your teaching style?
    - How do you assess your students?
    - For any of the curriculums that have a test at the end, do you feel like you are teaching to a test?
- Content
    - Do you like being able to develop your own curriculum or would you prefer a curriculum given to you?
    - Would you prefer specific standards for a computer science education curriculum, or a more high-level overview?
    - How do you feel about the multiple CS standards for teaching high school computer science?
    - What is something you think is missing from computer science education currently?
    - What do you see in the future of your computer science class?
    - Computer science is constantly changing, so how do you incorporate new topics and getting up to speed on those topics.
- Professional Development
    - Have you attended any professional development for any CS teaching? If so, what did you attend?
    - What is something you think is missing from computer science professional development?
- Have you heard of the CSTA standards?
    - Are you a CSTA member?
    - If so, what is your opinion on the standards?
    - Have you used any of them in to influence your curriculum?
    - Are you a CSTA member?
- K-12 CS framework
    - Have you used any of the K-12 CS frameworks in your curriculum?
- Computer Science principals and Computer Science A – AP classes
    - Do you teach any AP classes? If so, which ones?
    - Have you attend any professional learning courses offered by the College Board?
    - Computer Science A vs APCS principals. Does your school offer both? Are students able to take both courses?

    - What is your opinion on AP Computer Science A being similar to a first semester CS class, focusing on object orientated programming and problem solving and design using Java vs AP Computer Science principals teaching the underlying principals or computation.
- Computer Science IB classes
    - Do you teach IB classes, if so which ones?
    - What is your opinion on the IBCS standards
    - Does your school teach IBCS? If so, does it offer standard level, higher level, or both?
    - Have you attended any professional development provided by the IB, outside of what is required?
- If your school offers both APCS and IBCS, what do you think are the main differences between the two?
- Can you tell me some things about your education and background?

Appendix B – Principal Email

My name is Samantha Bewley and I am currently a master's student at Villanova University working toward my Master of Science in Computer Science. My thesis topic is current High School Computer Science Education and how teachers are navigating all the curriculum and standards currently being proposed and implemented.

A big part of my thesis is interviewing current High School Computer Science teachers. The interviews will either take place at the school, or over video conferencing software. The Institutional Review Board (IRB) has approved the questionnaire and research. The interviews are going to be guided by a questionnaire. I am emailing to ask if you will be able to put me in contact with your high school computer science teachers so that I may interview them.

Thank you!

Samantha Bewley

Appendix C – Teacher Email

My name is Samantha Bewley and I am currently working on my master's thesis at Villanova University. I am focusing on computer science education at the high school level, and the current standards and professional development. I was given your email by the principal of your school, {name of principal}.

I am interviewing high school teachers to get their opinions on the subject. If you are interested in participating, I will be able to meet you at your school at a date and time that works for you, or we can do the interview remotely. The interview will be about 30 minutes - 1 hour, I have received approval from the IRB for the interview and corresponding research.

Thank you,

Samantha Bewley

Appendix D – Interview Transcripts

So can you just tell me a little bit about your computer science courses that you teach?

I teach AP Computer Science A and AP CS Principles at [redacted] And since they're AP courses we are required to teach using an approved curriculum that the college board has approved.

So do you since you're given this approved curriculum that you have to teach to, do you prefer that like where you're given the curriculum or would you prefer to kind of develop your own and work on like you're own and what you think you would like to teach?

Well the CS A curriculum I used. I actually developed that using one of the approved curriculums as a guide. So I in effect was I guess writing it although I needed to adhere to the guidelines on what I needed to cover. The CS principles I did sign on to using an approved curriculum and I just found it really helpful to have those lessons that outline some examples of some assessments to use. And got… and you know and pacing.

What are some technologies that you use in your classroom to teach computer science?

Well I use IDE, an IDE. I use NetBeans for CS A and for principles I just use their online Google, I guess it's sites? The curriculum is all online. We Google sites and the linked videos and the linked lessons and the linked documents for activities.

Okay. So what would you say your teaching style is?

Good question. I guess… Style seems like such a broad category. What I try to do is I try to present my students with information I also try to keep them involved and get a sense of if they are struggling or not if they need some additional help. Where I can you know, where I can be of help or someone else in the class. Because often I find that students are just as knowledgeable and sometimes more knowledgeable than I am in certain areas. So I will use my students if I. Whenever I can.

So how do you assess your students?

I use for, I guess I use a similar type of assessment. They do get unit tests and a final exam. But I also use labwork quite often for assessing them in terms of completing a lab assignment.

The computer science AP curriculums, they both have that test at the end. Do you ever feel like you are teaching to that test instead of teaching like topics or ideas or stuff like that?

To some extent I do teach to make sure I've covered the concepts because the primary goal, at least my understanding is, that students are planning to take the test and I'd like them to do as well as they can, but I don't feel that there's a big disconnect between the exam at the end, the AP exam and any exams that I would give in class because not all my students tend to opt to take the AP exam so the ones that aren't are also taking a final exam and I do tend to design that using AP style questions.

CSTA is just one of the many different set of standards that they recommend for teaching computer science. How do you feel about having all these multiple standards instead of a few?

Right well I have looked at that. Well I don't. The standards that I use are the ones that the college board provides that are documented in the syllabus that I'm using. And I believe that because those courses specifically, particularly CS P is actually was created as… in combination with adhering to CSTA standards so I don't think that there's a big difference although I don't feel… I wouldn't describe me following the CSTA standards. I would describe my classes… that when I teach following the AP standards and the approved syllabus.

What is something that you think is currently missing from computer science education?

That's a good question. From the content of the courses themselves?

Just anything that could be missing.

Anything? I think that there's not a… since there's not a state requirement to take computer science that I think then that students are not required to take it. It's really reliant on their interest and even the move to accept a computer science

course as a math or science course credit in the state of Pennsylvania is a step in the right direction. But I don't feel that it's treated the same was as say a math course or core subjects or English. I think it should be… or a sciences because I think it's a required skill to succeed in life. So that's not really the content itself but I think the content is probably a great start.

What is something that you would like to see in the future of your computer science classes that you teach?

I'd like… I guess I would just like to see more students taking them. I'd like to see more courses offered not just two AP classes in my school.

So those are the only two that you teach. Are they also the only two that are offered at your school?

Mhm.

So also a big thing about computer science in the field is that it's constantly changing so how do you incorporate some new topics and how do you yourself get up to speed on those topics if you're going to incorporate them into your class?

Well I think in terms of new topics the CS Principles class is part of preparing for the explore task it's really actually helping all of us stay on top of what's, you know, new and different. Cause we go through a practice exercise. So I guess I along with some of my students are keeping on top of some things from that perspective. How do I stay on top of other things in terms. That are changing? I guess I've just always had an interest. I mean, teaching is not my first career. I used to work as a computer programmer for [redacted]. So I've had this. I guess I've had this passion for computer science and when I left that field and returned to teaching which… But I started out teaching math. But I brought in the computer science classes because I believe that it's such an important thing and there were lots of efforts going on and I had gone to some workshops at Carnegie Mellon about 10 years ago. So I think I myself just have this interest and I really try to support any students that have the interest as well. Cause we have a club that was newly started but it wasn't due to my support, but it was really on the students and encouraging some other students that might not have been… Might not have signed on unless somebody else had said you know this would be a great thing. You think you might be interested?

Have you attended any professional development for computer science teaching?

Well I'm currently the [redacted] for the Philly branch of CSTA. CSTA Philly. And so we have had… We've held some symposiums in the fall and the spring for any teachers in the area. So that is one type of professional development. I've also been an AP reader which I think is a fabulous professional development opportunity. I was for AP CS A 3 years ago, 2 years ago I was a table leader for Principles and this year I was a table leader for Principles again. I think that's a form of PD you can interact with other teachers. Computer science teachers are often the only computer science teacher in their school and it's one way for us to get networking… Get to know other teachers. And then I went to the mobile CS Principles PD that they had 2 summers ago, 3 summers ago. And before I started teaching CS A I went to AP summer institute which again was for CS A so that was the instructor helped us get ready to teach CS A.

Are your students able to take Computer Science A and AP Computer Science principles at your school?

Mhm.

Do they have to take them in a certain order or is it kinda just whatever they feel like?

Whatever they feel and whatever can fit into the schedule cause scheduling can sometimes be a problem.

Oh yes, that's the only reason I ever took a computer science class way back when actually cause it fit into my schedule.

Right so it could work in the reverse too. It could work where you're trying to figure out do I take computer science or do I take music at least that's the decision a lot of students in our school take because they tend to be scheduled at the same time.

How do you feel about the difference in the 2 AP Computer Science classes? Where AP Computer Science A kinda focuses more on the object oriented programming and the nitty gritty Java where AP computer science principles is kinda like the underlying like ideas of computer science and stuff like that.

If I… I know that they are suggested that they complement each other and that one is not a prereq for the other. But if I were to… If a student was asking me and if they had… they thought they might wanna take both I would suggest that they take CS P first but often some students if they don't take one of those until their senior year then they really… it's really up to them. In terms of…

And can you just tell me some things about your education, your background like and how you became a computer science teacher?

Well I originally graduated with a bachelor's degree in secondary ed math. And so but I never taught math at that time. Instead I joined a company as a programmer. While I was in school, I'm kinda dating myself, there was not a Comp Sci degree in my school. There were computer science classes but not a Comp Sci degree. And so I ended up working for this computer company. A small one first and then [redacted] later and while I was working there I ended up completing a master's in computer science while I was working full time as a programmer and I took some time… When I left [redacted] when my kids were young, my children were young to stay at home. Then I went back to school and I completed a master's in education and then decided that what I wanted to do instead of going back into industry was I wanted to teach and at the time I thought I wanted to teach math which I still teach. Most computer science teachers don't have enough of a course load to only teach computer science. So not only are they the only teacher, computer science teacher but they often have other things on their course load so I think with those 2 degrees I decided that I wanted to go into teaching, but it started out as math. But then it grew into adding computer science. So CS A I think I started teaching. I've been there. I taught 11 years. I probably started teaching CS A 8 years ago and CS P 2 years ago. Because it's only been out, launched, officially.

Awesome. Do you have any questions for me?

No, no I'm looking forward to...

So, can you just tell me a little bit about your computer science courses?

Yes. So, I never took any formal computer science courses because I went... I graduated from college in 1975 and computer science really was just at the cutting edge there and I was a math and physics major. My computer science

experience is all self-taught. After I taught for 5 years and I went to work for... to do software development. I learned FORTRAN on my own and I coded in FORTRAN for a long while and then they went to C and C++ so I migrated over to C and C++. While I was in industry I took courses at Penn in for a master's in computer science and I learned other languages and I guess my formal training would have been after the fact of actually writing code. So that's all. And then currently I teach Java but that's all self-taught.

And do you teach AP computer science here?

I do. I teach... we have 4 different classes here. We have intro programming which is... it's a semester class, half the year. C half the year, C++. Kids can sign up for either one. We have AP computer science A which is the Java class. We have the new computer science class which is AP computer science Principles that is any language that the kids want to write in. I show them a little bit of Java, a little bit of Python and a little bit of the drag and drop languages like Scratch and Snap and things like that. And we also have an advanced programming class that they have to know, they have to have gone through the AP Java class to take that.

So for these classes, how did you decide the curriculum you were going to use and what you were going to teach.

Well the AP... So the AP curriculum is dictated by the college board so we don't... there's no choice in that. The intro class has migrated over the years. We've always had an intro class and it's kinda changed language. For a while it was Pascal, sometimes it was, it was actually Java and that's kinda dictated. We felt that kids needed... We should give kids an opportunity to learn a different language and based on the fact that at some point the college board was Pascal, the AP language was Pascal, C++ or Java and what tools we have available. We like to use free tools if at all possible. So that kinda dictates. And then the advanced class is kinda what. It's more independent study so it depends how many kids are in and what they're interests are. Sometimes they work on Python projects, sometimes they do harder Java code.

Okay.What are some technologies that you use in your classroom?

You mean like the compilers and stuff?

Yea anything.

Okay so for Java we use Eclipse. It's a free download; it's perfect. For C and C++ we use CodeBlocks. I don't know if you've ever heard of CodeBlocks?

I haven't no.

It's not the greatest, but it works for what we need. Eclipse is really powerful and probably more powerful than we use for it, but it works really well. We have laptops, it's a one to school. Every student has their own laptop. I use, I'm not sure what other technology I use. I mean I use a SmartBoard to help instruct. I use my own laptop.

What would you say your teaching style is?

Well kids need to write code, do their own code and learn from their own mistakes. So I model code for them, like concepts, like nested loops and what it does. And then I give them a problem that's slightly different than the model I've just done but uses the concept, say of nested loops, so they have to think about it and so I'm not sure what that style would be. Maybe more student centered?

So then how do you assess your students?

I have rubrics for each of their programs. If it's an AP course, there... We have sets of tests that I've made up over the years that include multiple choice and the college board course in Java is they handwrite code. So I have made up problems mimicking that. In the other classes I don't have tests per se. We have projects in class and I grade them based on not necessarily getting the code to run in a class period but following the requirements even if they've missed semi colons and the thing won't compile they can still do well.

Okay, so for the curricula that have a test at the end like the AP Exam. Do you ever feel like you're teaching to that test instead of teaching concepts?

Well yes and no. That's not a definite. I do... You have... To teach Java you have to teach the concepts otherwise they're never going to be able to do well at all but we do... There's two of us here, we have enough classes for two teachers here. We both and we work in sync. But we teach like little tips to the test cause if kids are going to spend... I think the test costs like 95 dollars this past year and they want to do well. It looks good for colleges, sometimes it gives them some

credit. We give them tips on what they should do. For that, that would change a student getting a 4 to a 5 or a 3 to a 4 or even a 2 to a 3. So... but they can't... we follow their curriculum, but I wouldn't say we're teaching to the test in that respect. Does that make sense?

Do you like being able to develop your own curriculum or would you prefer the curriculum given to you like the AP computer science curriculum?

That's a good question. It doesn't matter. I mean I like the flexibility of my advanced programming class where I don't have to follow a fixed curriculum and if 3 students... Like this year I think there's 21 students in the class. If 5 of them want to work on a python project together, that's great. And if they want to spend 3 weeks on it as opposed to 2 where being in a forced curriculum that's tied to a high stakes test at the end you're stuck on a time line. So there's good and bad in both. I mean the high stakes test course you know, gives kids you know rigor and stuff but so does the other. Gives them more creativity. I don't think I prefer one or the other.

If you look online, there's multiple computer science standards, frameworks. All this stuff about teaching high school computer science. How do you feel about all these different options and all these different frameworks?

Well yea. Some of them are not really applicable. I mean a lot of them talk about social media, digital citizenship all that stuff. The AP curriculum, we do talk about... they have a little teeny piece of in their curriculum. Being good digital people. And in my intro class, in the other two classes. We have infused curriculums where we discuss how technology is helping and hurting people. I mean students need to be aware of what computers can do for them and yet their limitations and the dangers of them. So it would be nice if there were one set of standards. Everybody's trying to get to it. You know. So it's not quite... everyone's trying to weigh in. So it's good and bad. It's kind of confusing at the moment.

So what do you think is something that is currently missing from computer science education?

Well, curriculum wise, I guess we're not sure of what language is the best thing. I mean we're dictated by the college board AP - Java. In the intro... in the principles class, the computer science principles class, their other class. It's a

flexible language and it's more broad range so they're trying to address that issue there. What else might be missing? Women. Underrepresented groups are missing. We try to encourage people to take the course. It is difficult to think that way so I think what's missing in it... Starting it earlier, getting kids more exposed to it. That's what would be missing.

What would you like to see in the future of your computer science class?

I'd like to see… I'd like to see students, more students take the class. More students even if they don't take the AP test studies show that even if they take the class and do the rigor, they do better in college and experience you know a college type level course. I'd like to see in classes where I'm, we're doing, we the school district is doing the curriculum, I'd like to see flexibility. More students think of own things to do. More resources put toward it.

So on that note, since computer science is constantly changing and there's always new topics, always new things. How do you incorporate when students want to learn a new topic and then you yourself getting up to speed on that topic if you maybe aren't familiar with it?

Right, so no one can be an expert in everything in computer science for sure. I am not at all a network person or anything like that. So if a student was more interested in that in the more independent study class, the advanced computer class. I freely admit that I'm not an expert and will help them do research to do a project, but I can't instruct them on it any better than they can. In fact, for that matter those students that want to write harder Java code. I've never written Java code in a development environment, I've only ever self-taught for the course. So a lot of these kids, once they're in comp sci 2, they're further ahead in Java then I am and they'll ask me like can I help them debug a problem so we sit together and I freely admit. So I think just having the respect that I'm willing to admit that we don't know everything and that we're both learning together is a good thing. So I have no problem with that.

Awesome. Have you attended any professional development for computer science teaching?

Well yes. The college board offers week long courses in their curriculum. So I've attended theirs several times because the languages change. I've attended it for principles. I've graded the AP exam so that's kinda professional development. I've gone to... In fact I took a graduate course after we were going to Java at

Villanova, in Java, so I could be a better Java coder myself. So yea. I've taken that type of professional development.

What do you think is something that you think is currently missing from computer science professional development?

There's nothing in the schools that I'm aware of. It's all driven by outside forces. I would like to see more... like kinda hour of code. I would like to see more students participate even on a small scale basis. I would kinda like to see computer science in some fashion, I'm not sure what it would be, be required for every student. So I think that I'd like to see for professional development and I'm not sure right now in Pennsylvania... Math teachers tend to teach computer science. I'd like to see more in house professional development for math or science teachers to infuse that into their curriculum.

Have you heard of the CSTA standards for computer science?

Yes.

What is your opinion on those standards?

Neutral, you know. They're standards, you know I mean they're standards. You know again I'm not... Every standard, if you even talk about math every standard I'm not sure I'm 100 percent on board on but you know. I mean they're making a good attempt. They're maybe more knowledgeable on it than me.

Have you used any of the standards to influence your curriculum?

No not per se. I mean again 2 of the courses are dictated by the college board so that's... And I'm sure they're indirectly using standards. I haven't forced, well we're going to do this standard but like I said before we are infusing how computers are like...impact society, good and bad, economics, all that stuff so.

Since you teach both, or your school offers both Computer Science A and the Principles class, are students able to take both of them?

Yes.

And can they take them in any order or do they have to take one first?

No, they can take them in any order.

And then what is your opinion on the differences of the two classes? So how computer science A is Java and the nitty gritty coding and then Computer Science Principles is more like the underlying principles of computation and thinking how computer science works.

Right, so. You want my opinion on what the differences are?

Just how you think those differences come into play and do you like them?

Right, so well the college board wants the principles class to be more all inclusive. So anybody even like going into a business or English major would take it, be more aware of how computers are affecting their daily lives. So that's the purpose of that. The Java class, the A class is clearly for your techy, nerds that want to write code or be an engineer or some type of science major. That's the difference but so I unfortunately we're only at one class of principles but it's a good size. It's almost 30 but I'd like to make it so that we could get 2 people, sets of people in it. Yea, the problem with principles is because we don't... Not a problem but you get in there kids who have self-taught code. Kids who Comp Sci A. Kids who have never seen code in their life. It's a wide range of abilities but it's easily dealt with I think.

So that's all I had, do you have any questions for me?

No.

Works cited

"About AP Workshops | AP Central - The College Board." *AP Central*, The College Board, 5 Apr. 2018, apcentral.collegeboard.org/professional-development/workshops/about-workshops-summer-institutes.

"ABOUT CSTA." *Computer Science Teachers Association*, www.csteachers.org/page/About.

"AP Around the World." *AP Credit and Placement - AP Student - AP and College - AP Around the World*, The College Board, apstudent.collegeboard.org/creditandplacement/ap-and-college/ap-around-the-world.

"AP Computer Science A Course Description." *The College Board*, The College Board, 2014, media.collegeboard.com/digitalServices/pdf/ap/ap-computer-science-a-course-description.pdf.

"AP Computer Science Principles Course and Exam Description ." *The College Board*, The College Board, 2017, apcentral.collegeboard.org/pdf/ap-computer-science-principles-course-and-exam-description.pdf.

"Computer Science in DP." *International Baccalaureate®*, International Baccalaureate Organization, www.ibo.org/programmes/diploma-programme/curriculum/sciences/computer-science/.

"CSTA K-12 Computer Science Standards." *Computer Science Teachers Association*, sites.google.com/site/cstastandards/.

Daniel Heersink and Barbara M. Moskal. 2010. Measuring high school students' attitudes toward computing. In Proceedings of the 41st ACM technical symposium on Computer science education (SIGCSE '10). ACM, New York, NY, USA, 446-450. DOI: https://doi.org/10.1145/1734263.1734413

Darling-Hammond, L., Hyler, M. E., Gardner, M. (2017). Effective Teacher Professional Development (research brief). Palo Alto, CA: Learning Policy Institute.

Doukakis, Spyros & Psaltidou, A & Stavraki, A & Adamopoulos, N & Tsiotakis, Panagiotis & Stergou, S. (2010). Measuring the technological pedagogical content knowledge (TPACK) of in-service teachers of computer science who teach algorithms and programming in upper secondary education. Readings In Technology and Education: Proceedings of ICICTE 2010. 442-452.

"Enroll in AP Mentoring | AP Central - The College Board." *AP Central*, The College Board, 5 Mar. 2018, apcentral.collegeboard.org/professional-development/enroll-in-ap-mentoring.

Giannakos, Michail N., et al. "Investigating Teachers' Confidence on Technological Pedagogical and Content Knowledge: an Initial Validation of TPACK Scales in K-12 Computing Education Context." *Journal of Computers in Education*, vol. 2, no. 1, 2014, pp. 43–59., doi:10.1007/s40692-014-0024-8.

"Gov. Wolf Applauds Board of Education for Computer Science Standards." *Governor Tom Wolf*, Pennsylvania Government, 11 Jan. 2018, www.governor.pa.gov/governor-wolf-applauds-state-board-education-endorsement-computer-science-standards/.

Hendrickson, Katie. "A Vision for K–12 Computer Science." *k12cs.Org*, K-12 CS, k12cs.org/a-vision-for-k-12-computer-science/.

Hendrickson, Katie. "Teacher Development." *k12cs.Org*, K-12 CS, k12cs.org/teacher-development/.

Iborganization. "Benefits of IB for Students." *International Baccalaureate*®, International Baccalaureate Organization, www.ibo.org/benefits/benefits-for-students/.

Iborganization. "Diploma Programme." *International Baccalaureate*®, International Baccalaureate Organization, www.ibo.org/programmes/diploma-programme/.

"International Baccalaureate Diploma Programme Subject Brief ." *IBO*, International Baccalaureate Organization, ibo.org/globalassets/publications/recognition/4_computerhl.pdf.

"K–12 Computer Science Framework." *k12cs.Org*, Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and National Math and Science Initiative, k12cs.org/.

Mishra, Punya, and Matthew J. Koehler. "Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge." *Teachers College Record*, vol. 108, no. 6, June 2006, pp. 1017–1054., doi:10.1111/j.1467-9620.2006.00684.x.

Orenstein, Natalie. "New Law Supports Computer Science Education in Pennsylvania." Remake Learning, 14 Feb. 2017, remakelearning.org/blog/2017/02/14/new-law-supports-computer-science-education-in-pennsylvania/.

Pennsylvania Code § 57.31.

Popham, W. James. "Teaching to the Test?." *Educational Leadership*, vol. 58, no. 6, Mar. 2001, pp. 16–20.

"Professional Development | AP Central – The College Board." *AP Central*, The College Board, 16 Aug. 2017, apcentral.collegeboard.org/professional-development.

"Progression Chart - CSTA K-12 Computer Science Standards." *Computer Science Teachers Association*, sites.google.com/site/cstastandards/progression.

Sindberg, Laura. "Alone All Together- The Conundrum of Music Teacher Isolation and Connectedness." *Bulletin of the Council for Research in Music Education*, no. 189, 2011, pp. 7–22. *JSTOR*, JSTOR, www.jstor.org/stable/10.5406/bulcouresmusedu.189.0007.

"Work Toward College Success." *The Rewards – Explore AP – The College Board*, The College Board, apstudent.collegeboard.org/exploreap/the-rewards.