Worcester Polytechnic Institute
Digital WPI

Interactive Qualifying Projects (All Years) Interactive Qualifying Projects

March 2015

College Student Participation in K-12 Computer Science Education

Tony Garside
Worcester Polytechnic Institute

Xiaoman Xu
Worcester Polytechnic Institute

College Student Participation in K-12 Computer Science Education

An Interactive Qualifying Project Report

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

____

Tony Garside

____

Xiaoman Xu

Date:

March 4, 2015

Report Submitted to:

____

Professor Kathryn Fisler, Project Advisor

1.Computing Education

2.College Students

3.Teaching

Abstract

This report outlines recommendations for a program that would use college students as a resource to teach K-12 computing topics both inside and outside of Massachusetts classrooms. We also outline a recommendation for a conference to help push this program into fruition; organizing this conference could be a future WPI project. Our recommendations for this program are the result of research, interviews, and surveys, all of which are detailed in the report.

Acknowledgement

We would like to thank the staff at WPI STEM center, Shari Weaver and Lynn McElholm, for giving us much information about K-12 STEM and engineering education. We would also like to thank Nathaniel Granor at Technology Education And literacy in Schools program, for speaking with us about their program. In addition, we would like to thank WPI Engineering Ambassadors and the K-12 computing teachers to complete our online surveys. Finally, we would like to thank our advisor Kathryn Fisler for her guidance and patience throughout our project.

Table of Contents

1. Introduction

There are major unrecognized opportunities in K-12 computer science education; the Massachusetts Computer Attainment Network (MassCAN) has a mission to reform Massachusetts education to take advantage of them. MassCAN is a coalition of organizations motivated to bring computing education to Massachusetts K-12 classrooms. They are determined to not only make sure all Massachusetts students have a foundation in computing, but also to make Massachusetts the national leader in computing education. To accomplish this, MassCAN is focused on four major tasks to drive Massachusetts' computing education to the top of the country: educational standards, classroom curriculum, professional development for teachers who get involved, and public awareness of why computing education is so important.

By the year 2020, it is projected that there will be one million more computing jobs in the United States than computer science graduates to fill those positions (code.org, 2013). The demand for computing professionals is tremendous. In October 2013 there were 570,000 computing job openings nationwide, which is about 4 times the United States National average for all job sectors (The Conference Board, 2013). On top of this, computing jobs are easily one of the most top paying career choices, with an average salary of over $80,000 per year (U.S. Bureau of Labor Statistics, 2013). Even students who do not plan to become computer scientists benefit from a computing education. Computing classes are widely believed to help promote critical thinking skills as well as develop necessary computational skills. In a world where technology is rapidly advancing, there is a new demand for people to be technologically literate in all career fields.

The opportunities and benefits of a solid computer science education are not yet recognized by school districts across the United States. There is a major gap between what subjects are being taught in high schools and what skills are necessary to thrive in a job after the students graduate. Of all math and science AP tests taken in the United States, only 2% of them are for computer science (code.org, 2013). In the job market, however, 60% of jobs favor computing skills over all other math and science skills (code.org, 2013). Since 2005, the Computer Science Teachers Association (CSTA) has conducted surveys of high school teachers who define themselves as computer science, computer programming, or AP computer science teachers. In the most recent survey, conducted in 2013, of the 1286 respondents, 87% claim that they believe there are students who should be or would like to take computer science courses that their school offers, but do not (CSTA, 2013). The top ranking cause for this is that computer science classes are often counted as elective classes and therefore worth less credit. Students were less likely to choose computer science classes because they could not fit them into their schedules among their mandatory classes. This is a problem occurring in high school classrooms all across the United States; Computer science is not recognized as a math or a science, but instead an optional elective class.

Currently there are not enough educators in the United States that are qualified to teach computing in public schools. According to Computing in the Core, an organization advocating K-12 computer science education, high school introductory computing classes have decreased 17 percent from 2005 (Computing in the core, n.d.). It is not feasible to send teachers already active in classrooms back to school to learn about teaching computer science, however, this is a great opportunity for outside help to get involved. It would be expensive to attempt to hire professionals from industry to teach computer science to younger kids. College students, on the other hand, are an untapped resource when it comes to public school education. College students are a much better fit, as they have more flexible time to invest, are just as capable of teaching basic computing topics, and they have a smaller age gap between themselves and K-12 students which may allow younger students to have an easier time relating to them. College students could play a significant role in the early stages of transforming the Massachusetts education system. Say a public school teacher is interested in teaching computing in their classroom, but he or she lacks a formal computer science education. College students could be used as a resource to overcome this barrier through making themselves available to teachers in classrooms to do things such as answer student questions or provide technical demonstrations. This simplifies work for the teacher in addition to being a great opportunity for college students involved to get experience in teaching, public speaking, and solidifying their foundational knowledge of computing.

We are helping MassCAN to design a program that allows college students to get involved in computer science education with K-12 students and teachers. There are many potential ways in which college students might get involved. We have created a program infrastructure to engage school districts in computer science education through enabling college students to participate in K-12 classrooms. We also have to gathered useful resources to help student volunteers teach K-12 students computer science. In addition to this, we researched how we can make college students available to conduct activities and demos to class field trips from local districts. Because public awareness is a key aspect to promoting changes in education, we researched how we can get college students involved in public awareness of computer science education. Lastly, we outlined a recommendation for a conference to be held at WPI. This conference should foster discussions among college students, professors, and volunteers to help bring a college student aided K-12 computer science education program into actualization in Massachusetts.

Most college students are not well equipped with the tools necessary to teach in a classroom: they require some training first. Because college students should only be expected to teach K-12 students with the classroom teacher present, college students would not be required to learn about disciplinary issues with K-12 students. Instead, the college student training should be composed of standards for teaching sciences to children. College students participating in the program should be educated on how to speak appropriately to the age of students that they are working with. They should be trained to stimulate the minds of the students they're working with using methods such as Socratic questioning, rather than just reciting information. Depending on how in-depth each age group training program is, college students may have to specialize in a particular age group.

The task of putting college students in classrooms will require time and resources from schools and teachers. Bringing K-12 students to college hosted events may be a better option for school districts just looking to explore the idea. Instead of sending college students off to K-12 classrooms, a hefty commitment, it may be easier to bring K-12 students to college hosted events for those just looking for a small amount of computer science exposure. This would be less of a commitment for everyone involved. College students would only have to dedicate a few hours one day of the week to the program and would require far less training than if they were being sent to a classroom. K-12 teachers and school districts would have an opportunity to get exposed to what a computer science program for kids might look like, without committing nearly as much time and resources. While being less of a commitment, this option is still a fantastic way to promote computer science to K-12 students.

Our interactive qualifying project aims to accomplish two main objectives. Our first goal is to develop a prototype/recommendation for a program that uses college students as a resource to foster interest and educate K-12 students in computing concepts. To accomplish this we must first understand the perspectives of both college students and K-12 teachers to lay out a foundation for how this program will be conducted. We must also understand how to teach young students and be knowledgeable of best practices when it comes to teaching K-12 students math and science. To gather this information, we have conducted surveys, interviews, and observed successful programs that function similarly to the program that we would like to implement. Our second goal is to outline a recommendation for a conference for future iterations of this IQP to conduct to stimulate interest and discussion among local college students and faculty to help bring this program into realization. We aim to accomplish this through documenting discussion topic suggestions based on the research we have conducted.

2. Background Data Research

Before designing a proper program that uses college students as a resource to help K-12 teachers with computing education, we must first learn college students' and teachers' interests in this program. Furthermore, we want to locate the opportunity where college students can come to help. After gathering enough data by surveys or from other published research, we want to shape our program by looking at some existing similar programs to have a general idea of what the program should be like.

2.1 College Students' Interests and Participation

College students were K-12 students, so we are interested when they decided to be a computer science major to determine if we need to pay more attention on some K-12 grade level. On the other hand, college students plays an important role in our program concepts since we want to them to help teach, so we need to know or increase their interest in this program to make sure this program can work out.

2.1.1 An IQP From 2013-2014

An IQP in 2013 – 2014 focused heavily on understanding the perspective of college students around their potential participation in a K-12 program. They conducted a survey of 89 WPI computer science students to gather general data about students who may be interested in participating in such a program. Of the 89 students that responded to the survey, roughly 70% claimed that they had decided to pursue computer science as their major when they were in high school. When asked about their interests in computing, 51 respondents credited their choice of studying computer science to their interest in programming. Of the 89 students who participated in the survey, 26 of them stated that they would be interested in working with pre-college students in an effort to generate K-12 student interest and involvement in computing. Of those respondents, there was an equal ratio of male to female students with 10 of them being first year students, 7 being sophomores, 5 juniors, and 4 seniors.

From this information we can conclude a few things. It appears that a majority of student interest in computing tends to develop at the high school level. Programming appears to be the driving factor for generating that interest. Based on the data that was gathered, the demographic of students who may be interested in participating in a K-12 outreach program is diverse, having a balanced gender ratio and being spread amongst students of all levels of experience.

2.1.2 WPI Engineering Ambassadors Surveys

To better understand college students' interests and actual participation in working with K-12 student to help teaching, we conducted a survey of WPI Engineering Ambassadors, who give K-12 students presentations in engineering topics (for more details, see 2.4.2 Interview with Lynn McElholm). Both students who responded to the survey give two presentations to K-12 students per term, which takes about 8 hours, more than their once-a-term requirement. In terms of time commitment, they have weekly meetings that last 1.5 - 2 hours, in addition to their personal preparation time, which, according to one of them, took around five hours a term. Summing up, they spend 30 hours per term on this program in total. As we have learned from the STEM center, this program requires new Engineering Ambassadors to be trained before they can start their work. Both of the survey takers thought the training program well prepared them to fulfill the job requirement. One participant described the training program as robust training. The other one admitted that the training gave a fundamental style of presentation, though in fact they learned more from practice. Both of the students enjoy working as an Engineering Ambassador. One felt happy that they serve as an encouragement for young kids to learn engineering and therefore can make huge impact on those students' lives. The other one said that besides interacting with kids, they Engineering Ambassadors can meet other hardworking students in college, also gaining leadership, communication, teamwork and presentation skills through this program. The two Engineering Ambassadors who took the survey both believed that getting paid and internships, plus community service hours are sufficient enough incentive to get college students like them more involved in outreach programs for kids.

2.1.3 Summary

These two survey results may help us to understand college students' interest and where we should work on to develop the program. The following are the conclusions:

- In most cases, computing major students decided their major in high school because of their interest in programming. This indicates that high school is an important time interval to develop students' interests in computing field.
- There are a number of college students showing interests to work with pre-college students but most of them are freshmen and sophomore. The reason might be that juniors and seniors are busier or interested in other fields.
- College students who participate in K-12 engineering education program enjoy their work and are happy to encourage younger kids to learn their major field topics. This indicates the potential opportunity to engage college students in K-12 computing education.
- Compensation such as salary and internship could interest more college students to participate. We may take these factors into consideration when we are ready to build our program.

2.2 K-12 Teachers' Perspective

K-12 computing teacher is another important role in our program because they take charge of K-12 classrooms. We want to know what kind of challenges they have while teaching their students, and how they think college students may help.

2.2.1 Computer Science Teachers Association (CSTA) National Secondary School Computer Science Survey

CSTA has conducted surveys of national secondary school in computer science education every two years since 2005. The following is part of the survey data.

From the survey data of 2013, of 1286 respondents,

- 74% survey participants' school offered introductory Computer Science courses. However, only 40% made the credit earned by the course(s) as Computer Science or Computing Credit. 46% regarded it as Tech Credit. 23% made it Business Credit, and the remaining 4% designed as Science Credit.
- 87% of teachers thought that there should be students who would like to take CS course but do not, and the reason ranking first was "no room in timetable".
- The greatest challenges that teachers thought was lack of students' interest and enrollment, and then lack of staff support or interest. Besides that, teachers regarded rapidly changing technology and difficult subject matter as great challenges.
- The greatest professional needs that teacher perceived were "time for training", and then "training opportunities".
- The respondents believed that the most efficient methods for delivering professional development to CS teachers were workshops and seminars.

From the data above, we learn that a number of K-12 schools do not know how to deal with computing courses because they are not compulsory for students. For the same reason, K-12 students would not like or do not have time to take computing courses. College students are not capable of solving these problems, but if they could interest more kids to learn computer science, and make more people to be aware of the importance of computer science education in K-12 level, it may help increase people's awareness of making certain standard of K-12 education. At the same time, most K-12 teachers indicate that they would like to take training for their teaching career yet do not have time or opportunities. Teachers need training because they want to build up their professional background. This implies the possibilities that college students can go to K-12 classroom to help K-12 teachers teach computer science with the knowledge in their major field.

2.2.2 K-12 Computing Teachers' Survey

We attempted to conduct a survey of K-12 computing teachers to learn more about their challenges when they try to explain computing topics. However, since we can reach few K-12 level computing teachers, we only got four valid responses. All of the respondents teach in high school from different states.

One of the teachers has taught computing topics for three years in California. Being a math major originally, he[1] was mostly self-taught in computer science, with only one introduction to C++ course taken in college. He thinks that it is not easy to get students combine ideas. For example, students may understand how *if* statement and *while* loop works separately, but they can be confused when putting *while* loop inside an *if* statement. He feels that college students can help assist his students with coding and answer questions that he does not have enough time to answer in class. As for college students' preparation, this teacher thinks that college students need to learn how to teach and how to lead K-12 students to find answers by asking questions, rather than simply write code for them.

A teacher who works in New Jersey has a Bachelor of Arts Degree in Computer Science and has taught computing topics for ten years. He wants college students to share their experience and expectations in his classroom, be patient and able to explain topics. Beyond the classroom, he thinks college students can help promote the importance to know the computer in workplace. He does not elaborate on teaching challenges. Another teacher who has taught computing topics in Rhode Island more than 10 years also does not think there is particular challenge for him in explaining topics, rather it is a greater challenge to have his administrators and colleagues understand what they do. He thinks that college student should mostly be prepared to share their excitement and enthusiasm about their chosen field of study. He was majored in marketing and took one programming class as an undergraduate and has a Master of Business Administration Degree. The rest programming knowledge was learned online or self-taught.

[1] All "he" in section 2.2.2 means "he or she".

Another teacher seems to misunderstand most of the questions but he did express the ideas that college students can “become the bridge between the word of programming and the rest of the world” and “Computing, logic and programming is everywhere, make it accessible, make it fun”. We interpret this as saying he wants college students to promote the general awareness of the importance of computing. This teacher thinks college students can look into projects that excite younger students, sharing the experience with next generation. This teacher is teaching in Massachusetts. He has taught four years as part of his Advanced Engineering and Robotics class. He has Bachelor of Science in Industrial Arts, Master of Arts in Occupational Education. He said he has a strong background in Mechanics and self-taught in programming.

An interesting thing is that of the four respondents, the relatively less experienced teacher tends to want more help of college students to answer students’ questions and explain concepts, while another two teachers who have taught at least ten years want college students to share their experience with K-12 students. They agree that college students should be able to explain topics. One of the much more experienced teacher thinks that college students can also help promote public awareness. All of these suggest that college students could have a lot of effects on K-12 computing education.

2.2.3 Teachers' Professional Development

During our research we studied some professional development programs to see K-12 teachers' needs in their teaching career. As students, it's difficult for us to understand how a K-12 computer science program would be run from a teacher's perspective. Teacher professional development programs help us understand what is most important to the teachers when teaching computer science. However, most of what we found was specific curricula for teachers who already has known how to teach, such as AP Computer Science A Professional Development (APCS) and Exploring Computer Science (ECS) Professional Development, rather than training for new teachers. The following are some examples of teacher professional development.

AP Computer Science A Professional Development

The AP Computer Science A professional development program aims to teach new and experienced AP teachers how to prepare their students for the AP Computer Science A exam. After attending the workshop, participants will be capable of:

- Aligning their classroom instruction with the goals of the AP Computer Science A course
- Identifying the skills and knowledge that the exam will assess as well as identifying the tasks and topics for which students may require more help in preparation for the exam
- Drafting a syllabus that meets the curricular requirements for the course

This program in particular has a curriculum module that provides strategies for introducing and practicing recursion. Recursion has consistently been the lowest-scoring topic on the AP Computer Science A exam.

More information on the AP Computer Science A Professional Development program can be found here: http://professionals.collegeboard.com/prof-dev/workshops/math/ap-comp-sci

Exploring Computer Science (ECS) Professional Development

The Exploring Computer Science professional development program aims to educate teachers about computer science through three major ideas (referred to as pillars): computer science content/concepts, inquiry, and equity.

The program model begins with a week long summer session to introduce the instructional philosophy of the curriculum and content from the early units. During the academic year, participants must attend quarterly Saturday workshops to cover the remaining units and cover their three "pillars" units. In addition to this ECS coaches visit the classrooms if teachers to help provide individualized support. ECS's webpage provides a link to 2014's summer agenda, which gives some insight on what kind of topics are taught in the first session of their program. Lastly, ECS seeks to provide deepening of computer science content and pedagogy past the 2 year mark of participating in their program.

More information on the Exploring Computer Science Professional Development program can be found here: http://www.exploringcs.org/teacher-support

2.2.4 Barriers and Supports to Implement Computer Science (BASICS) -- Outlier Research & Evaluation at the University of Chicago

Outlier Research & Evaluation at the University of Chicago is carrying out a two and a half year research study to examine the status of implementation of an introductory Computer Science course, Exploring Computer Science (ECS). One of their research questions is that "What supports and barriers affect ESC use and its potential endurance?" The research team conducted a survey in Chicago and Washington, DC in Spring 2014 to ask high school Computer Science teachers to identify the three biggest supports for and barriers to their computer science classes. Two tables of the survey results are attached in page 14 and page 15, "BARRIERS to Teaching Introductory Computer Science" and "SUPPORTS to Teaching Introductory Computer Science".

Based on the data shown on these two tables, we conclude that college students can play a part in K-12 Computer Science education. In the "BARRIERS" table, the barrier #1 of "Teachers Themselves" section is teachers' own trepidation because of lacking specialized knowledge and experience in ECS teaching style. In "supports" table, lots of teachers indicate that they need to have professional development sessions or look for other help to be familiar with teaching materials. The interaction between teachers themselves can also help with course teaching. These show that there is a great opportunity for college students to come to help because they can provide their professional knowledge as a support for K-12 teachers. As for the barrier that parents may not understand and encourage their children to learn CS, college students can hold some public events such as presentations, showing and explaining their projects to increase public awareness. K-12 students themselves are another problem because they may not have much interest or not feel confident in learning computers. Some students want more direct hands-on activities using computers. For this point, we can bring K-12 students to college campus to have tours, hands-on projects and demonstrations, which are led by college student because they know better about college project experience and can give K-12 students a more direct feeling about how it is like to learn computer science. For the rest of barriers components in the table such as funding, budgets and technical difficulties, though college student cannot improve the teaching condition directly, it is reasonable for them to know about these situations to better understand K-12 education problem. On the other hand, college students can help interest K-12 students to learn computer science and hold more events to increase public awareness so that more people will pay attention to the problem such like funding.

BARRIERS to Teaching Introductory Computer Science (with *Exploring Computer Science* Materials)

Key
- Red: Barrier #1
- Purple: Barrier #2
- Green: Barrier #3

Teachers Themselves
- **Self-Efficacy**
  - [My] own trepidation
- **Experience**
  - Need basic knowledge of robots and web design
  - Lack experience in ECS teaching style
- **Management**
  - Time management
  - Classroom Management

Families
- Parents
- Getting parents to understand what the course is about
- I don't feel that parents know enough or encourage their children enough about CS

Students
- **Behaviors**
  - Students' attendance
  - [Lack of] willingness to engage and give the material a chance
  - Students' lack of focus
  - Students' discipline
  - Disruptive/disinterested students
- **Self-Perceptions**
  - According to students, they don't feel they are good at computers
  - Students' self esteem
- **Needs**
  - [Lack of] support
  - Many students want more direct hands-on activities using computers
  - Teaching students with extreme special needs/cases (many students are in and out of jail)

School/ Organizational
- **Resources/Space**
  - Lack of funding
  - Lack of funding
  - [Lack of] space
  - Technical difficulties
  - [Lack of] support for labs
  - Spending any money
  - Lack of classroom space
- **Scheduling/ Electives**
  - CS is not a requirement
  - Not enough elective spots open in student schedules
  - ECS becomes a dumping ground since it's an elective class
  - School schedules (A/B schedule)
  - Students have limited elective choices
- **Communication**
  - Students report that they don't know what the class is about
  - Getting counselors to understand what the course is about
  - Lack of promoting CS classes
- **Collaboration**
  - Insufficient collaboration with other teachers

District and School Support
- **Administrative**
  - School budgets were tight and principals are often faced with choices between supporting test scores and providing CS classes
  - [Lack of] change in system
  - No administrative support
- **Other Teachers**
  - Other teachers do not recognize CS as a core class and often use my class for time for extras like practicing for a play
  - Teachers at my school do not value CS
  - Other teachers sometimes express disdain for CS

ECS Program Components
- **Lessons**
  - Some lessons are too low-level for my students' capabilities
  - Sorting lesson
  - Need materials necessary to teach ECS
  - CSS lesson
  - Robotics unit should include lessons for Finch because it is more affordable than LegoMindstorm
- **Model**
  - Does not meet the needs of our other departments
  - ECS model doesn't fit with district priorities and evaluation
  - guidelines for teachers
  - Assumes students are well versed using MS applications
  - ECS model sometimes assumes too much independence from students that they will lack

Professional Development
- PD Sessions

None
- None
- None
- None

In Spring, 2014, 24 Computer Science teachers in Chicago and Washington, DC completed a questionnaire that, among other things, asked them to identify the three biggest supports for and barriers to their computer science classes. All of the teachers were using *Exploring Computer Science* (ECS) instructional materials.

This visual represents what teachers said were their biggest supports to teaching computer science (unedited).Teachers were asked to provide a text response for "Barrier 1" "Barrier 2" and "Barrier 3." Nineteen of the 24 teachers provided at least one response.

This material is based upon work supported by the National Science Foundation under Grant Number CNS - 1339256

[BASICS] BARRIERS AND SUPPORTS TO IMPLEMENTING COMPUTER SCIENCE

outlier RESEARCH & EVALUATION CEMSE | UNIVERSITY OF CHICAGO

SUPPORTS to Teaching Introductory Computer Science (with *Exploring Computer Science* Materials)

Key
- Red: Support #1
- Purple: Support #2
- Green: Support #3

- **Professional Development**
  - **Sessions**
    - ECS professional development
    - ECS professional development
    - ECS/ToC [Taste of Computing] training
    - PD sessions
    - Regular PDs
    - Summer PD
    - Taste of Computing
    - ECS professional development
  - **Facilitators**
    - The ECS/ToC team helped me to feel confident and comfortable with the materials
    - ECS PD leaders
    - ToC presenters
- **Teachers**
  - **At My School**
    - Teachers in my department
    - Fellow teachers at my school
    - Other teacher who works with me
    - Fellow teachers in CTE [Career and Technical Education]
    - Other teachers in my department
    - Fellow Teachers
  - **At Other Schools**
    - Other ECS teachers
    - Other ToC teachers
    - Regular interactions with other ECS teachers
    - CS teachers in ToC CSTA [Computer Science Teachers Association] community
    - Teachers at other schools
    - Former teacher colleagues at other schools
    - Networking with other teachers
    - Teachers to talk to
    - CS teachers in external community: NCWIT [National Center for Women & Information Technology] & SIGCSE [Special Interest Group on Computer Science Education]
    - Professional community
- **Resources**
  - **Online Community**
    - Online community
    - ECS teacher community online
    - Online CS groups
  - **Instructional**
    - Materials (technical and non-technical)
    - Resources provided by the district
    - Pre-created objectives
    - Books and reference materials for ALL STUDENTS
    - Resources
    - ECS materials
    - ECS materials
    - Online Resources
    - Assessments
  - **Stakeholders**
    - Co-teaching (at least once a week for the whole year) by a computer scientist from AAAS [American Association for the Advancement of Science]
  - **Financial and Physical**
    - Money
    - Space
- **School and District/ Organizational**
  - **Administrative**
    - Program manager
    - School administrators and computing staff
    - District administrators
    - Administrative support
    - Principal
    - The administration does provide me with feedback about teaching in general but doesn't understand CS and seems a little intimidated by it
  - **CTE**
    - CTE Department
    - CTE
- **Students**
  - Student support
  - Tech savvy students
- **Parents**
  - Parental support
  - The parents have expressed that they are happy with what I am doing

In Spring, 2014, 24 Computer Science teachers in Chicago and Washington, DC completed a questionnaire that, among other things, asked them to identify the three biggest supports for and barriers to their computer science classes. All of the teachers were using *Exploring Computer Science* (ECS) instructional materials.

This visual represents what teachers said were their biggest supports to teaching computer science (unedited).Teachers were asked to provide a text response for "Support 1" "Support 2" and "Support 3." Nineteen of the 24 teachers provided at least one response.

This material is based upon work supported by the National Science Foundation under Grant Number CNS - 1339256

[BASICS] BARRIERS AND SUPPORTS TO IMPLEMENTING COMPUTER SCIENCE

outlier

2.2.5 Summary

Our research shows that, as we have supposed, a number of K-12 computing teachers do not have enough professional background and experience and thus have trouble in explaining topics while teaching, so they would like to take training and participate in workshops. This proves that there is a chance for college students going to K-12 classroom to help teach with their professional knowledge. Nevertheless, we have additional findings:

- Different teachers may have different needs from college students. Some teachers may want college students to share project or study experience with K-12 students. We can try to match K-12 teachers' needs with participating college students' personal strengths when we build our program, because some college students may be good at explaining concepts while some others more like to present their projects.
- Besides lacking background, another big challenge for K-12 teachers is that K-12 students may not take computing courses because these courses are not compulsory and students are not interested at all. Therefore, aside from helping teach, college students can do presentations on general introduction to Computer Science to increase K-12 students' interests.
- A lot of people do not realize the importance of computing education. K-12 computing teachers' colleagues may not understand computing teachers' work and K-12 students' parents may not support their children learn computing. Hence, beyond the classroom, college students can help with promoting public awareness of the significance of computing.
- There are many other problems that college students cannot solve such as lack of funding and enough technical equipment in K-12 schools. However, college students should learn about these problems to better understand the difficulty in K-12 computing education and thus be more prepared before they going to help teach.

2.3 Existing Programs That Use College Students in K-12 Education

To design such a program that engages college students in K-12 computing education, we need to know what college students should do and provide in K-12 classrooms, after-school activities, and what kind training they may need to work effectively to help K-12 teachers with K-12 computer science education. Fortunately, a couple of universities, colleges and organizations have realized the possibility of using college students as a resource to promote K-12 education, and therefore provide us some good examples of both sending college students to K-12 classrooms and bringing K-12 students to colleges.

2.3.1 Engineering Ambassador at WPI

Engineering Ambassador at WPI is a program that uses college students to motivate K-12 students to learn engineering by giving engineering topics related presentations. For more details, see section 2.1.2 and 2.4.2.

2.3.2 Student Teacher Outreach Mentorship Program (STOMP)

Student Teacher Outreach Mentorship Program (STOMP), founded in 2001 at Tufts University, provides an opportunity for college students, high school students and industry employees to develop engineering education in K-12 schools. For more details, see section 3.1.

2.3.3 STARS Computing Corps

Different from the way that universities and colleges engage their own students, a national community of regional partnerships – STARS Computing Corps – provides a platform, connecting universities and K-12 schools, through which students at participating organizations are able to engage in K-12 education. Its main program, STARS Leadership Corps, can be either a curricular or co-curricular service-learning program, which encourages student-led regional engagement. For more details, see section 3.1.

2.3.4 Technology Education and Literacy in Schools

Technology Education And Literacy in Schools (TEALS) recruits volunteers in computing industry and places them in high school classes to help local teachers satisfy students' demand in acquiring professional knowledge. Working with classroom teachers, some TEALS volunteers are teaching assistants (TA), focusing on grading assignments, while others prepare and deliver introductory or AP lessons. Before they teach, all volunteers are trained by TEALS during the summer. Each volunteer is expected to spend 20 hours on online and in-person training sessions and another 30 hours working with their teaching team and doing homework. After training, all volunteers are expected to attend approximately 50% of all class sessions throughout the year. Eventually, classroom teachers take over the CS course after getting enough support from the volunteers so that the school can maintain and develop a sustainable CS program independently. For more details, see section 2.4.3.

2.3.5 Outreach Program of University of Minnesota, College of Science and Engineering

The outreach program of University of Minnesota College of Science and Engineering is a good example of attracting K-12 kids to its campus. The K-12 outreach program holds activities in K-12 schools and community programs to foster kids' interests in science and engineering. With the kids being attracted, the university hosts summer camps or tours on campus or in research laboratories, allowing them to get further knowledge in STEM fields. During the camps or the tours, student volunteers at the university would help with all kinds of activities, which usually lasts for one hour. These activities includes mentoring robotics teams, leading lab tours, presenting students' projects and researches, etc.

2.3.6 Outreach and K-12 programs of University of South Carolina, College of Engineering and Computing

University of South Carolina (USC) also demonstrates the feature bringing K-12 kids to college campus well. To get K-12 school students prepared for college study in STEM and develop prospective students as well, USC's College of Engineering and Computing offers two types of tours: Big Friday Sessions and Daily College of Engineering Tours. Current students at USC lead the Daily College of Engineering tours during the weekdays. They also attend the Q&A panels in Big Friday Sessions, which offer more comprehensive overviews of the College of Engineering and Computing than the daily tours do.

Another example is Enhanced Learning Experience (ELE), which gives high school students chances to work directly with college professors and current college students by attending special classes. ELE aims to interest high school students in computing and engineering through hands-on projects.

We can learn from these activities to conduct small and quick demonstrations led by college students for K-12 schools.

2.3.7 K-12 Programs of University of Washington, College of Engineering

University of Washington's K-12 programs offers multiple programs where K-12 students can get involved in engineering with college students. Much like WPI, the University of Washington has their own Engineering Ambassadors program. Their Engineering Ambassadors visit K-12 classrooms to provide presentations on engineering topics or career opportunities within engineering, hands on activities or facilitate demonstrations, background on the different types of engineering, college preparation and general application process. Meanwhile, DawgBytes is their program that aims to introduce K-12 students and teachers to computer science and engineering. It hosts programming competitions, computing open house, summer camps, and more to get K-12 student involved in computer science and recruits college students as volunteers to help these events around. Even though their "Engineering Ambassadors" do not go to school to teach, which is different with the program we would like to build, this K-12 program can be a good example to both have college students to go to K-12 schools and provide small activities for K-12 school students on campus during after school time.

2.4 Interviews

As the research we have conducted so far, we realize that the situation of STEM or engineering education in K-12 schools may be similar with computer science. To further understand K-12 teachers' perspective of working with college students, we have interviews with Shari Weaver and Lynn McElholm, who are staff at WPI's STEM Education center. We also have an interview with Nathaniel Granor, one of the manager of TEALS (see section 2.3.4), to learn how the training program organizes as a reference.

2.4.1 Interview with Shari Weaver

Our interview with Shari Weaver, a K-12 outreach staff member of WPI's STEM Education center, gave us a lot of useful information about how college students can be used in K-12 Education. We think one of the biggest challenges for our program will be preparing college students to teach K-12 students. If a program where college students work directly with K-12 students were to be implemented, Shari recommends that there should be special training for college students for working with different student age groups built into the program. She directed us to Massachusetts mathematics educational standards documents as well as a set of standards called Next Generation Science Standards to use as resources to understand what science topics to teach different K-12 age groups. Shari suggests we use programs such as WPI's Engineering Ambassadors (see section 2.3.1), Tech girls, and the American Institute of Aeronautics and Astronautics K-12 program as additional successful outreach programs to aid our research. Shari believes that integrating computing education into K-12 classrooms is a rising opportunity. Schools are being pushed to invest into computer labs for standardized testing; when testing is not taking place, they could be used for computing education. All that is missing is someone to teach it.

Our interview with Shari also gave us a good perspective of the challenges that K-12 teachers face when teaching computing subjects to their students. Shari believes that the main reasons that K-12 teachers do not want to teach computing are because of lack of experience and fear caused by that lack of experience. Most elementary school teachers lack a background in computer science, as they mainly focus their education in teaching kids basic reading, writing, and math skills. Even teachers who may be capable of teaching computing still require resources and material to teach it. Currently there are very few computer science standards and activities that are readily accessible to K-12 students. Shari recommended that we look into currently existing standards such as the Massachusetts curriculum Framework for Mathematics which could serve as pseudo standards until there is a set that is formally agreed upon.

When speaking with Shari, we also gathered information from the perspective of college students and the challenges they may face when teaching computing subjects to K-12 students. A major factor that will impact the success of a college student to K-12 outreach program is how well prepared the college students are to teach and convey information to younger students. Shari recommends that college students involved in such a program understand how to work well with kids. She informed us that there are currently very few existing programs that that focus on this sort of preparation and that the training program may have to be built into the outreach program itself. Shari warned us that it can be difficult to encourage young students to take interest in STEM subjects. She recommends that perhaps the approach of putting a college student in the classroom could be very beneficial, as the college student could serve as a role model to motivate and inspire students.

2.4.2 Interview with Lynn McElholm

Our interview with Lynn McElholm, the director of Engineering Ambassadors, gave us more details about this program. According to Lynn, this program model was first built at Pennsylvania State University (Penn State) in 2009. Their college students were going out to high school to try to reach underrepresented groups with engineering education. United Technology Company (UTC) saw how the program worked and then decided to help spread this program. Hence sponsored by UTC, the Engineering Ambassador Network was established in Pennsylvania State

University, University of Connecticut, Worcester Polytechnic Institute (WPI), and Rochester Polytechnic Institute (RPI). This program in WPI started with 10 Engineering Ambassadors. Now, the Engineering Ambassador team has extended to 17 students. If they are able to, Engineering Ambassadors students go to middle schools and high schools to give engineering related presentations, but due to WPI students' tight schedules, they usually host K-12 students on campus to show college experience and foster hands-on projects.

Lynn explained to us that one of the reasons kids are less interested in engineering is because of the image engineering currently has. People often associate Engineers with math, physics and introversion, whereas with something like doctors, people associate them with benevolence and selflessness. The mission of Engineering Ambassadors is to convey the ideas that engineering is interesting and it will make a better world to the young group instead. For this reason, Lynn indicated that the most important thing in recruitment process is training. The training program is divided into three parts and uses two books, *The Craft of Scientific Presentations* and *Changing the conversation*: it begins at the end of September, giving participating students the first exposure to what will be used in their future job. In term break between A and B term, which is the mid to the end of October, the students have their second training, where they have more time to get used to what they should do and practice presentations. When B term starts, they learn where they have done well and where they should work on more in weekly meetings, by the videos of their presentations. Overall, the training program gives Engineering Ambassadors the overview to the whole program, the presentation style they use and the way they talk to kids.

Here is the link to Engineering Ambassador Network

http://www.engr.psu.edu/ambassadors/partnersUniversity.html

2.4.3 Interview with Nathaniel Granor

Our interview with Nathaniel Granor, the East Regional Manager of TEALS, gave us insight on the inner workings of how the TEALS program works. TEALS shares fundamental similarities with the program that we are building a recommendation for. We mainly focused our questions about the TEALS volunteer training program. Nathaniel explained to us that as the TEALS program began to grow in its early stages, scalability became a problem. One thing that must be taken into account for is that what may work for a small program centralized program may not be viable as the program grows and thrives. TEALS began as a very small program reaching out to only 12 students in its first year. Now, only five years later, they're reaching out to thousands of students across 18 states. TEALS has made reforms to the format of their training process to account for this. Classes are organized nationwide for volunteers to attend and lectures and material are posted on the TEALS partnership Dashboard for volunteers to review or keep up to date with if they cannot make it to in person gatherings.

Nathaniel told us about the challenges volunteers ran into when teaching computing to students. One of the more common problems that he has observed is the managing students of different experience levels with technology. In a classroom, students come from all different levels of knowledge and experience with computing. It's difficult to make sure that students with strong backgrounds in computing are challenged while keeping less experienced students engaged. Another common problem Nathaniel expressed was that often inexperienced volunteers struggle with making sure their students are paying attention and retaining the information that they are teaching. Within the TEALS curriculum there are lessons on how to recognize when a student loses interest and how to help regain students' attention. Lastly, Nathaniel expressed the importance of using Socratic questioning as a method to engage students without directly giving students the answers to their problems, he explained that it's important to remember that teaching is not the act of delivering information.

Nathaniel gave us some recommendations for a program that uses college students to teach K-12 rather than industry professionals. He reminded us that the structure of college student lectures is very different from how K-12 classrooms are formatted. This is something college students need to be aware of when they work with younger students. What may work for college students may not be a suitable way to teach younger students. Pre-college students tend have shorter attention spans, less devotion, and are accustom to a completely different style of being taught than the lecture style that college students are used to. There's also much less of an age gap between college students and K-12 students than in the TEALS format where industry professionals teach the students. Nathaniel cautioned us to make sure that college students in classrooms have a firm understanding that they must take on leadership roles; they are K-12 students' teachers, not their friends. Lastly, Nathaniel recommended that the program focus its attention on college students' time management. College students must be conscious of scheduling and allocate appropriate amounts of time for classroom commitments accounting for unexpected issues such as traffic or difficulties setting up equipment for class.

2.5 Outreach for Underrepresented Minorities and Women

It's important to remember that not all students have the benefit of growing up with technology in their household. These students, typically underrepresented minorities, are less likely to be interested in computing topics because they are either not exposed to them at an early enough age or they simply don't understand what computing actually is. Those that do choose to pursue an education in computer science are put at a disadvantage due to their lack of experience growing up. When designing an outreach program we must be conscious of the culture that we create. We must be careful in our design to ensure that all groups of people feel welcome.

We looked into the programs run by National Center for Women in Technology (NCWIT), which is an organization that specializes in making technology workforces and education more accessible to women and minorities. NCWIT provides resources for best practices teaching young students while being inclusive to all groups of people. NCWIT has articles such as "Top 10 Ways to Engage Underrepresented Students in Computing" and "Top 10 Ways to Increase Girls' Participation in Computing Competitions" which could prove to be useful resources for teaching women and minorities in computer science. These and other resources can be found on their website at http://www.ncwit.org/.

2.6 Summary

Up to this point, we learn that there are a number of college students interested in working with K-12 students, and many computing teachers express that they need assistance in classroom to help answer questions because of lack of professional knowledge or experience. Therefore it is very possible that college students become a good resource for K-12 computing teachers to help explain topics or share college experience with K-12 students. Furthermore, by looking at existing programs that use college students in K-12 education, we conclude that besides sending college students to K-12 classroom, it is also feasible and probably much easier to bring kids to college campus to learn Computer Science. College students can give presentations and lead activities for visiting K-12 students. However, we realize that college students must be well prepared before they teach. From our survey results and interviews with staff at WPI STEM Education center and one of the program managers at TEALS, we learn that basically college students should be able to explain topics and know how to connect with kids. Thus the training for college students turns out to be necessary.

3. Program Design

In the program we want to design, we would like to improve Massachusetts K-12 computing education by making college students a good resource, including going to K-12 classrooms to help K-12 teachers with designing course or answering students' questions, and giving visiting K-12 students presentations, demonstrations or hands-on projects. At the same time, we would like to interest K-12 students in learning computing science. Before we can implement the program, it is important to know what the program should be like, including program coordination and other details.

3.1 Ideas to Adopt from STOMP and STARS

We made a comparison between the program we want to implement and similar programs to have a general impression on our program design.

STOMP at Tufts University pairs every two participating Tufts students with a K-12 teacher to assist in the integration of engineering in the school curriculum and bring new technologies and creative activities to the classroom. With the help of classroom teachers, Tufts students design and teach an innovative engineering curriculum, which consists of 8-10 weekly sessions, each an hour long. Since fall 2014, STOMP has engaged 59 students. They work 5 - 10 hours each week, including preparation and in-class time. To have college students prepared for K-12 teaching, STOMP offers a couple of trainings in educational technologies and teaching skills. STOMPers are also required to attend weekly development meetings. We want to design training programs and weekly meetings as well to better prepare our college students. But it is noticeable that STOMPers are really going to teach engineering courses. They have to design and submit a unit outline at the beginning of each semester and then update it each STOMP day. We discuss college students teaching format in next section 3.2.1. STOMP also has a nice table "Grade Profiles" in their STOMP Handbook (Tufts University's Center for Engineering Educational Outreach, 2014), which states what college students should notice and do in K-12 classroom based on kids' age. We want to construct such kind materials for our college students in our training program because teaching can be a lot different through out K-12 level. In general, this program is a good example for sending college students to provide their professional skills in K-12 classrooms.

STARS Corps can serve as a good example to show how we can implement the program. The universities or colleges that join STARS Corps design their own projects and programs, which basically aim to encourage and educate pre-college students to prepare them for computing major in college. This is generally implemented as curricular or co-curricular program, for example, either a course bearing credits, a student organization, or honor program etc. We can build our program, especially the training program, as a curricular program, too. The time commitment for a participating student is around 5 to 10 hours a week for on average of 2 semesters. Depending on the design of the project, college students may do K-14 outreach, mentoring, tutoring, pair-learning, internships, and community service. This program also builds a Digital Library, which provides resources about their projects online. These activities and resources give us a good sight on what college students can do in this program, but we may want mainly focus on teaching activities. Some of these projects focus on fostering college students for their future computing career. In another word, the main goal of STARS Corps projects is to expand the number of people in computing field. Their college students may work with K-12 schools, yet not many related to directly offer help in the classroom. For example, their project topics also include Career Development and Research Experience for Undergraduates. But they do have a few good examples, such as teaching or training robotics and social media integration on elementary and middle school students. We can learn from these projects to eventually design our own.

3.2 Recommendation and Resources

After having a general impression on what the program should be like, we will now talk about some recommendation and resources to implement different parts of the program.

3.2.1 Recommendations for program that puts college students into K-12 classroom

The following recommendations could help college students have a good sense about what and how they should do in K-12 classroom, preparing them with K-12 teaching and some social issues.

1. **Design a program that trains college students to work with K-12 students**

It is noteworthy that college students may not be familiar with K-12 education. Thus, it is necessary to provide training before we send them to K-12 classroom. The training sessions mainly focus on teaching context and technique, and the training could be designed as a one or two weeks long course or a weekly seminar through a semester for the students in the program. It could also be set up online if participating college students are at many different colleges.

a. **Context -- Understanding K-12 classroom**

- **College students need to be aware of the difference between college and K-12 classroom**

From our interview with Nathaniel, we have learned that there can be a huge difference between the style of college lecture and K-12 class, such that younger kids tend to have shorter attention span and less devotion. If college students tried to explain topics as the way to which they accustom in college, it might not work out with kids. Therefore we want to help our college students understand how college and K-12 classrooms function differently. For example, we can send college students to several field trips to let them watch how it works in K-12 classroom. Such kind activities may only need one or two days in the training week for college students to observe and conclude.

b. **Teaching technique**

- **Design a curriculum to teach college students to connect with and teach K-12 students**

After college students have the first impression on the style of K-12 class, it is still important that they know how to get along with K-12 students, such as using proper words to explain terminology to kids and giving praises and questions to keep kids' interest in learning. We can design a curriculum that provides college students some resources, such as videos and readings, to connect with K-12 students. This may take one or two training sessions for college students to learn and apply.

Junior Achievement is a program that puts volunteers to K-12 classroom to foster K-12 students' work-readiness, entrepreneurship and financial literacy skills. This program posts a set of training videos that help volunteers to work with kids when delivering classes in elementary and secondary school, such as how to deal with kids' questions that may not relate to the class and what to do to keep the class on task. The videos also include some strategies to use when working with students in a diverse environment, and suggestions when leading an activity. They can be found through the following link.

https://www.juniorachievement.org/web/ja-usa/training-videos

There is a Delaware course to prepare students to go into K-12 classrooms and teach. It trains students both in CS professional skills and teaching skills. In the reading list of this course, there is a report on teaching skills, *Eight Thing Skilled Teachers Think, Say, and Do* (Larry Ferlazzo, October 2012). It demonstrates how teachers shall think, say and do while teaching kids. For example, when thinking of teaching style, teachers shall consider authoritative way rather than authoritarian way, since the former will make students more likely to respect the teachers' authority and cooperate; when teaching a class, teachers shall give positive messages and say "please" and "thanks" because praises and respect can encourage students' interest and confidence in learning. These techniques can be useful for college students to gain a great sense that what is needed to be aware of while teaching. The Delaware course syllabus can be found at http://www.eecis.udel.edu/~pollock/syllabus-14F.pdf

- **Watch sample computing course videos and tutorials to help with teaching technique**

If college students were going to teach Computer Science in K-12 classroom, they may lack teaching techniques. Sample teaching videos and tutorials can easily give them visual impressions that how to teach young kids. We can let college students watch a couple of this kind of videos and then conclude what they learn or what technique they see in common.

Code.org directs users to other websites that contain programming tutorials on its "Go Beyond One Hour" module. Some of these tutorials are video-based courses teaching a specific programming language, such as JavaScript and Python. An example would be "Intro to JS: Drawing & Animation". This course contains teaching videos on subjects like "Drawing basics", "Coloring", "Animation basics" and so on. College students can watch some of these videos to learn how to explain a programming concept and conclude what should be aware of. This may take two training sessions.

http://code.org/learn/beyond

https://www.khanacademy.org/computing/computer-programming/programming

On twitter, @CSTeachingTips has plenty of resources about very specific teaching technique such as "Use graphics when possible to keep students engaged" and "Explain that arrays work like a dresser of clothing to help students understand this abstract data structure". These tips are like conducting small teaching activities via text. On their official website, the teaching tips could be filtered by category such as algorithm, data structures and recursion, etc. College students may look at these tips to help them prepare to teach a specific topic in Computer Science.

Here are the links:

https://twitter.com/csteachingtips

http://csteachingtips.org/

- **Design a curriculum to teach college students to give presentations properly**

College students can go to K-12 classroom to just give a presentation on computing topics instead of teaching a real course. To do this they need to learn how to give presentations properly and keep practice. Regarding this, we refer to two books that are used in the training of WPI Engineering Ambassador, *The Craft of Scientific Presentations* (Michael Alley, 2013) and *Changing the conversation* by National Academy of Engineering (2008). These two books address on how to be a good presenter of science and engineering and showing people a different way to understand engineering such that learning engineering can make the world better, instead that learning engineering needs excellent math and physics. Though these two books focus on presenting engineering, K-12 computing education faces similar problems -- people usually think computer science is too hard to learn, rather than it can change the world. We can adapt general ideas from these two books and change the topic to computer science.

College students can be nervous with little experience in their first presentation or teaching, so we can give them more practice opportunity in their training. For example, college students can give a sample presentation while being videotaped on a general computer science topic such as *if* statement, and later on we can do peer-review or critique on these presentations to improve their teaching skills. It may take one or two training session to introduce the proper presentation styles and three days for college students to practice and get used to it.

- **NCWITS tips**

Besides the proper way to connect with K-12 students, college students still need to be aware of dealing with underrepresented groups and girls in computing fields. Because of growing environment, such groups may tend to have much less interest in computing than others. NCWITS (see section 2.5 for more details) provides some programs and tips we can look at to help college students concern this problem, such as "Top 10 Ways to Engage Underrepresented Students in Computing" and "Top 10 Ways to Increase Girls' Participation in Computing Competitions". For instance, they suggest using curricula like game design and digital media to engage students who are new to computing, and make promotional materials represent and appeal to girls. This may only take one training session. The following are the links:

http://www.ncwit.org/resources/top-10-ways-engage-underrepresented-students-computing/top-10-ways-engage-underrepresented

http://www.ncwit.org/resources/top-10-ways-increase-girls-participation-computing-competitions/top-10-ways-increase-girls

http://www.ncwit.org/resources?field_audiences_tid%5B%5D=1

2. College students teaching format: Teacher or Teaching Assistant (TA)

There are basically two options for college students in K-12 classroom: a. Teach a computing course as a K-12 teacher usually does; b. Help K-12 teachers answer K-12 students' questions and grade the homework as a Teaching Assistant and give presentations. If we are going to develop such a program, we need to know which role college students should take responsibility.

TEALS (see section 2.3.4 and 2.4.3 for details) and STOMP (see section 2.3.2 and 3.1 for details) are two programs whose fellows actually teach in K-12 classrooms. Apart from being a teacher, volunteers at TEALS can also be a TA who helps with grading homework, along with slightly less time commitment than a teacher. However, volunteers at TEALS are all high tech professionals. They can teach as early as 7:30am to fit in their normal business schedule. Students at STOMP design their own courses and teach 8 – 10 classes each semester. Finally K-12 teachers will be familiar with engineering, while college students can enhance communication and leadership skills.

"Teacher" model sounds great. However, we must take college students' daily schedule into consideration. Learning how to teach and get along with K-12 kids can be much time-consuming. TEALS volunteers are expected to attend approximately 50% of all class session over the year. STOMPers need 5 – 10 hours per week to prepare their lessons, excluding the time traveling to K-12 school. It is great that college students can gain an awareness of the education system through such experience, but we need much more concern to decide whether this model is appropriate. After all, they are still college students, and thus have to focus on their academic courses.

College students can still teach computing topics by giving presentations, which requires much less time commitment than teaching a whole course. WPI Engineering Ambassadors (see section 2.3.1 and 2.4.2) give presentations both in K-12 classroom and on college campus to teach basic engineering concepts. This means while college students going to K-12 classroom to help teach, K-12 students can be brought to college campus to learn. The time commitment of engineering ambassadors is around 4 – 5 hours per week. This model will not take too much time of college students while teaching K-12 students computing knowledge, yet just because so college students might not have so much communication and connection with K-12 teachers. Thus it is hard to say if college students can really help new teachers to be familiar with professional knowledge.

When implementing this program, we may assign different roles such as teachers or TA to college students according to their time flexibility, but we suggest doing some additional research to find out the best teaching format of college students.

3.2.2 Recommendations for program that brings K-12 students to college campus

- **Have resources and materials for on campus demonstrations and activities**

On campus events will require activities and demonstrations. It is important to have fun, engaging activities to hold students' attention. Demonstrations should aim to be short and include hands-on work to keep kids involved.

Sample demonstrations and activities:

Below we have compiled a list of computing activities that college students may find suitable to employ in K-12 classrooms.

Coding Related:

- **Code.org** offers a number of activities to introduce young kids to programming. On code.org's activity webpage you can expect to find tutorials feature simplified coding languages where kids follow step by step procedures to make simple games and animations.

Ages: Code.org has activities for students of all ages, ranging from 4 to 104 years old.

Time commitment: Code.org has activities designed to take 1 hour to complete as well as up to 20 hour long courses.

Link: http://code.org/learn

- **Scratch** is a simple, drag and drop programming language that can be used to demonstrate programming concepts. Typically scratch is used to program simple animations and games.

  Ages: Scratch is designed to students ranging in age from 8 to 16 years old.

  Time commitment: Scratch activities are perfect for quick activities, some as short as 10 to 15 minutes long. Generally you can expect Scratch projects to be about an hour long.

  Sample activities: http://www.teach-ict.com/programming/scratch/scratch_home.htm

  Official webpage: http://scratch.mit.edu/

- **Alice** is a simple, easy to use, programming environment that can help teach children programming concepts. Alice is like more advanced version of Scratch, featuring drag and drop coding with a higher level of customizability.

  Ages: Alice activities are typically designed for students ranging from 10 to 18 years old.

  Time commitment: Alice activities are typically longer, ranging in time from less than an hour to yearlong academic courses.

  Official webpage: http://www.alice.org/index.php

  Sample activities: http://www.cs.duke.edu/csed/alice/aliceInSchools/lessonPlans/

  http://www.aliceprogramming.net/materials.html

Computer literacy:

- **Kid's computer lab** is an educational website full of computer guides and tutorials to teach general computer literacy. Kid's computer lab has topics such as general desktop navigation, document formatting, and internet research skills.

  Ages: Kids computer lab has activities for kids ranging from ages 5 to 11 years old.

  Time commitment: Most tutorials provided by the Kids computer lab organization are short, students could potentially complete multiple tutorials in one hour.

Official webpage: http://www.kidscomputerlab.org/

- **McGraw-Hill Companies** has a webpage dedicated to basic computer literacy lessons for kids. The lessons range from learning how to use a computer mouse to how to make a basic webpage.

Ages: These computer lessons are designed for kids ranging from ages 5 to 12 years old.

Time commitment: Each full lesson is designed to be done in about an hour.

Official webpage:

http://treasures.macmillanmh.com/e/teachers/resources/computer-literacy-lessons

Computing without computers:

- **Computer science unplugged** contains computing activities that do not require access to a computer. This is a useful resource to teach computing in schools that cannot afford computers for every student. An assortment of basic computing topics can be taught using computer science unplugged.

Ages: Computer science unplugged activities are designed for students ranging from age 5 to 18 years old.

Time commitment: Computer science unplugged activities are designed to be done in sessions ranging from 1 to 3 hours in length.

Official webpage: http://csunplugged.com

3.2.3 Logistical Recommendations

- **A method for teachers or K-12 schools to provide feedback about their experience with the program**

In programs like WPI's Engineering Ambassadors, there are ways participating students can reflect on what they excel at and what they need to improve upon. It is important to have the tangible examples of things that students participating in the program can discuss and improve upon. We recommend making a feedback form available to teachers and K-12 schools to allow them to communicate their experience with the program. This could be done through either emailing survey forms to schools who are involved periodically, or leaving a feedback form on the program's webpage. The responses to feedback forms should be used to foster discussions during weekly meetings to allow students to reflect on areas that they can improve themselves when working with K-12 students.

1. **Design weekly meetings**

From studying other successful programs, we have determined that it is beneficial to hold weekly meetings. Programs like Engineering Ambassadors (see section 2.3.1) use weekly meetings for everything from administrative announcements to practice sessions. We recommend having weekly meetings integrated into this potential computing program. Depending on how much information needs to be conveyed will determine the duration of meetings. We recommend keeping meetings to no more than 1-2 hours in length one time every week or once every two weeks.

2. **Announcements and upcoming events**

Meetings are a great opportunity to gather all participating students in one place for administrative purposes. Meetings should begin with announcements and participants should be given handouts with announcement information. Announcements may include upcoming K-12 tours and activities, reminders for optional school seminars or events, as well as the dates and times for future meetings. Announcements should also be posted on the programs webpage for students to reference.

3. Critique student presentation

In our student training recommendation we suggested that program administrators video tape participating college students performing sample presentations. We recommend that time is reserved in some or all meetings for students to review the video of their presentations. This will give students the opportunity to critique themselves and others allowing them to improve their presentation skills. Students should be encouraged to focus on both the positive and negative aspects of their presentations; try to keep a healthy, balanced discussion of what went well and what did not.

4. Student discussions

We recommend setting aside time in meetings to facilitate student discussions about their experiences while working with K-12 students. This time could be used for students to ask questions about situations that arose while working with younger students that they were unsure of how to handle, how they handled the situation, and how they could have handled the situation better (if possible). It is important to have student based discussions to give students the opportunity to ask questions and express any concerns that they may have.

5. Discussions about teachers' feedback

In our recommendation we include a section about having a method for teachers to provide feedback to the college students on their performance working with K-12 students. Every few meetings should have time dedicated to reviewing the feedback that students received. This is very important for students to do. Students should be aware that above all, the goal of this program is to be helpful to K-12 teachers and their students. Sometimes it may not be clear to the participating college students of whether or not they are actually accomplishing this. Honest feedback from K-12 teachers can provide insight to participating college students. Feedback should be taken seriously and students should be encouraged to improve themselves based on criticisms provided by the K-12 teachers that they work with.

6. Practice lessons/materials

Meeting times are a good opportunity for participants to spend time practice giving lessons and activities. It is important that students thoroughly understand the information contained in the lessons that are being provided to K-12 students so that they are prepared to help and answer questions. Students should be encouraged to run through practice lessons on their own time, however meetings give them the opportunity to do so while working with their peers.

- **Program Coordination**

  1. **One or more people to run and organize the program**

Organizing a statewide K-12 computing program that pairs college students with K-12 classrooms is without a doubt a challenging feat. The program will require at least one person dedicated to leading the program and making administrative decisions. The program leader should be responsible for running program meetings, communicating with teachers and school districts, as well as keeping college students organized and motivated.

  2. **Central method of communication**

When running a program that intends to connect college students with K-12 students and teachers, there must be a way for K-12 schools to get in contact with the head of the program. There are many potential ways of doing this, but no matter how it's done, there needs to be some sort of central entity in charge of communication.

School level coordination

Most currently functioning programs connect the barrier between K-12 schools and college students through maintaining a contact form on their webpage. We recommend that a successful computing program design and maintain a form to allow teachers and parents to contact the head of the program. It may be useful to include questions about the demographic of the students looking to participate in the program. Information such as the age, gender, racial diversity, and the general income of the students can help program administrators plan the type of activities in preparation of their visit.

Examples of contact forms used by other organizations:

University of South Carolina College of Engineering and Computing Big Friday program: http://www.engr.sc.edu/tours/tour-registration_form-BIGfridays.html

University of Washington Engineering Ambassadors program:

http://www.engr.washington.edu/ambassadors/index.html

University of Minnesota Chemists in the Classroom program:

http://www.chem.umn.edu/outreach/about.html

Statewide level Coordination

In conjunction with a school level coordination, it may make sense to also organize communication on a statewide level. Have the head of the program contact teachers and districts to arrange the program.

- **Method to incentivize college students to participate**

College students have commitments outside of participating in this program and will expect to be compensated for their time. Last year's IQP, "Program to Increase K-12 Interest in Computer Science", contains a document outlining funding options for college students (see Program to Increase K-12 Interest in Computer Science, Appendix H).

- **Public Awareness**

One final thing to consider when organizing a statewide K-12 computing program is attracting the attention of the public. As stated in our research above, the need for computer scientists is growing rapidly. In order to fill those positions, we need parents, teachers, and school districts to acknowledge how big computing is becoming and begin taking steps to advance computing education in Massachusetts. In addition to this, there still exist obstacles that stand in the way of formal computing education for young students.

As stated in earlier sections, there exists misinformation about what computer science actually is. Young students tend to be less interested in computing because they assume computing jobs are boring typing jobs rather than teamwork-oriented work in fields that make major impacts on the world. Another big problem that computing education currently faces is that most people outside of computing do not realize the importance of it. The public needs to recognize that this is a massive field that is projected to grow rapidly over the next few decades at least. One final problem that we found in our research is that while K-12 schools may have the drive to advance their computing education, they may not be capable of organizing proper classes simply because they lack funding and equipment. While this sort of is not the responsibility of college students to fix, it is good to be aware of these problems. Perhaps growing awareness focused on computing can help play a part in generating funding to allow less wealthy schools to start promoting computing programs.

There are many ways college students can help in promoting public awareness, such as holding information booths at technical events for kids or organizing information sessions for parents and teachers. We recommend that college students participating in a K-12 computing program work with their mentors to come up with ways that they can spread information about the growing opportunities in computing with K-12 parents and teachers.

4. Next Step: Conference

Up to this point what we have accomplished is defined a recommendation for a potential program that uses college students as a resource to teach K-12 students computing topics. So we have a plan outlined, the question is: what is the next step to bringing this program into fruition? From here, we believe the next logical step to push this program forward is to start having discussions, generate interest, and further refine details. In order to facilitate these discussions for a K-12 computer science program, we feel it may be beneficial to organize a conference. We feel having a conference is a good exercise to begin talking about the program, flesh out details, and foster interest from potential students, teachers, and school that may be interested in participating in a K-12 computing program.

4.1 Who should attend the conference?

Having a conference is a major time commitment, not just for those organizing it, but also for those who choose to attend. We want to make sure that the information presented at the conference is useful and that we don't waste anyone's time. The audience of the conference must be chosen carefully to facilitate the planned discussions. We brainstormed ideas for who could attend this conference and what it could be about. It is up to students of future iterations of this project to actually decide who should attend based on the discussions they feel would be most beneficial in planning the program. The following list explores some ideas of who could be invited to the conference as well as how each particular audience will influence the discussions that will take place.

1. **Conference attended by K-12 teachers and school district officials**

- Conference could be about generating interest for K-12 teachers and school districts to explore the possibility of implementing a K-12 computer science outreach program.
- Conference could be about presenting a rough idea of a K-12 outreach program to teachers and school district officials. The conference could be used as a discussion tool to gather useful data to aid in the design of our K-12 outreach program. This is especially useful because it allows the program to be formed from the perspective of the teacher, which is something we as students have limited access to.

2. Conference attended by college students

- Conference could be about generating interest for college students to participate in a computer science outreach program.
- Conference could be more discussion based, a description of the program could be presented and questions from college students could be answered afterwards.
- Conference could be used to brainstorm and reform the K-12 outreach program from the perspective of college students.

4.2 What should be discussed at the conference?

There's still a lot of details to this program that must be discussed it can come anywhere close to actualization. It is up to the students of future iterations of this program to actually decide what is discussed at the conference, however, we have put together some examples of topics that we feel would be the most beneficial to discuss.

- **Program administration**

  One major part of the program that needs to be discussed and refined is the actual details of who runs the program and how. Should the program be run be one person central to all colleges across Massachusetts? Should every college have a program leader than is responsible for their own college students? What is the responsibility of the program leader(s)? How should he/she/they communicate between each other and K-12 schools? The program will need a strong set of guidelines that decide all of this information.

- **Publicizing the organization**

  Obviously the program can never come into actualization if no one wants to participate. One of the benefits of holding a conference is that it provides an opportunity to advertise the program to college students and faculty that may be interested in participating in the program. College students should be invited to the conference and there should be discussions held focused on how more interest in the program can be generated.

- **Program format**

  We outline some ideas for how the program can be formatted in our recommendation section, however these are far from finalized ideas. We recommend you use our suggestions as a baseline and build off of them to create solid guidelines for the format that the program follows.

Bibliography

Code.org Infographic Source Data. (2013). Retrieved October 15, 2014, from https://docs.google.com/document/d/1gySkItxiJn_vwb8HIIKNXqen184mRtzDX12cux0ZgZk/pub

May 2013 National Occupational Employment and Wage Estimates. (2014, April 1). Retrieved October 15, 2014.

Make Computer Science in K-12 count. (n.d.). Retrieved October 15, 2014, from http://code.org/files/convince_your_school_or_state.pdf

CSTA National Secondary School Computer Science Survey. (2013). Retrieved October 15, 2014, from https://csta.acm.org/Research/sub/Projects/ResearchFiles/CSTASurvey2013Comp.pdf

BARRIERS to Teaching Introductory Computer Science (n.d.). Retrieved from https://s3.amazonaws.com/cemse/basics/images/BASICS_Barriers_Large_edited.pdf

SUPPORTS to Teaching Introducory Computer Science (n.d.). Retrieved from https://s3.amazonaws.com/cemse/basics/images/BASICS_Supports_Large4.pdf

Tufts STOMP | Student Teacher Outreach Mentorship Program. Retrieved October 15, 2014, from http://sites.tufts.edu/stomp/

STARS Computing Corps (n.d.). Corps Process and Components | STARS Computing Corps. Retrieved October 15, 2014 from http://unccslc.starsalliance.org/corps-process-and-components

TEALS | Computer Science in Every High School. Retrieved October 15, 2014, from http://www.tealsk12.org/

K-12 Outreach: University of Minnesota, Twin Cities: College of Science & Engineering. Retrieved October 15, 2014, from http://cse.umn.edu/k12/index.php

Outreach and K-12 Programs: College of Engineering and Computing: University of South Carolina. Retrieved October 15, 2014, from http://www.engr.sc.edu/outreach/

K-12 Programs UW College of Engineering. Retrieved October 15, 2014, from http://www.engr.washington.edu/alumcomm/k12.html

"Facts and Resources." *Computing in the Core*. N.p., n.d. Web. 26 Feb. 2015, from http://www.computinginthecore.org/facts-resources

Alley, Michael (2013). *The Craft of Scientific Presentations: Critical Steps to Succeed and Critical Errors to Avoid (Ed. 2)*. New York: Springer.

Changing the Conversation: Messages for Improving Public Understanding of Engineering. Washington, D.C.: National Academies, 2008.

(2014) *STOMP Handbook*. For Fellows: Documents. Retrieved from http://sites.tufts.edu/stompactivitydatabase/files/2014/08/STOMP-Handbook.pdf

Ferlazzo, Larry. *Eight Things Skilled Teachers Think, Say, and Do.* Educational Leadership: Students Who Challenge Us: Eight Things Skilled Teachers Think, Say, and Do. Association for Supervision and Curriculum Development, Oct. 2012. Retrieved from http://www.ascd.org/publications/educational-leadership/oct12/vol70/num02/Eight-Things-Skilled-Teachers-Think,-Say,-and-Do.aspx.

Appendix A: WPI Engineering Ambassadors Survey Questions:

1. As an Engineering Ambassador, how often do you give presentations or do other outreach work?

2. What is your time commitment working as an Engineering Ambassador?

3. Do you feel the training program provided by Engineering Ambassadors was/is enough preparation for what you are required to do? Please explain why or why not.

4. What do you enjoy about working as an Engineering Ambassador?

5. What do you feel is a good enough incentive to get college students like yourself more involved in outreach programs for kids? College credit? Money? Other?

6. How is the Engineering Ambassadors training program structured? Is it a one-time program at the beginning of the year? Are there meetings throughout the year?

Appendix B: K-12 Computing Teachers Survey Questions:

1. Which state are your teaching?

2. Which level of K-12 are you teaching? (E.g. elementary, middle school, grades 7-8, etc.)

3. How many years have you taught computing topics?

4. As a computing teacher, what kind of background in computing did you have prior to teaching? (I.e. major in college, self-taught, working in computing area etc.)

5. What kind of challenges do you have when trying to explain computing topics to your student?

6. In what ways might college students be a useful resource in your classroom (either for you or for your students)?

7. If college students were going to a K-12 classroom to help, what should they do to prepare themselves?

8. Beyond the classroom, in what ways could college students help with K-12 computing education?

Appendix C: Interview Questions

Interview with Shari Weaver:

1. If a program involving college students teaching STEM to K-12 kids was implemented, what kind of training or preparation do you think a college student would require?

2. What kind of challenges do teachers face when teaching STEM topics that college students may be able to help with?

3. Do you know any events or projects that college student could be involved in to promote the awareness for STEM education?

4. Do you know of any resources that could help us learn more about teaching STEM to younger kids?

5. Do you know any similar programs that use college students as resources to teach STEM?

6. Do you know anyone at WPI currently running a program that uses college students to teach K-12 students in other fields?

Interview with Lynn McElholm:

1. What issues has Engineering Ambassadors had sustaining the student interest?

2. How is the Engineering Ambassadors training program structured?

3. What is the recruitment process of Engineering Ambassadors?

4. What is the purpose of weekly Engineering Ambassadors meetings? How are they conducted? What is discussed? How crucial are the meetings to the program?

Interview with Nathaniel Granor:

1. How is the training program structured?

2. What is the hard part of the training? What part do you think is effective?

3. What skills is hard to teach in classroom?

4. What full time teachers most need?

5. Do you have any other advice for our program?

6. Do you have any pointers to the materials used in the classroom?