March 2014

Program to Increase K-12 Interest in Computer Science

Adam John Degenhardt
Worcester Polytechnic Institute

Anthony John Tamburri
Worcester Polytechnic Institute

Benjamin Michael Ouimet
Worcester Polytechnic Institute

Brittany Ann Owens
Worcester Polytechnic Institute

Owen Green West
Worcester Polytechnic Institute

Program to Increase k-12 interest in Computer Science

Owen West, Adam Degenhardt, Brittany Owens, AJ Tamburri, Ben Ouimet

March 9, 2014

A Major 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

Owen West   Adam Degenhardt

Brittany Owens   AJ Tamburri

Ben Ouimet

Date: March 2014

Approved:

Professor David Finkel, Advisor

Table of Contents

Part I: Overview

Part I A: Abstract

This report outlines a program that aims to involve Computer Science students in the high-school and middle-school classroom and thereby foster interest and involvement in the Computer Science industry. The final design of the program is the result of research, interviews, and program testing, all of which are detailed in the report.

Part I B: Mission Statement

Our mission is to increase participation in the Computer Science field by fostering interest at a middle- and high-school level. We took a three-pronged approach to getting college students involved in the grade school classroom. First, we designed a presentation that college students will use to kick off their involvement in schools and then presented it at the Advanced Math and Science Academy in Marlboro to get student feedback. Second, we built the foundation for a student to get involved; we designed two slightly different sets of guidelines: one for high school and one for middle school. We have also provided interested Computer Science majors with the resources and paperwork they need to fill out in order to be vetted for work in the classroom. Third, we designed a tour that Computer Science majors will take their students on at the end of their time in the classroom. All three of these aspects of the program are applicable to any grade school or college which wishes to implement our program.

Part I C: Three Section Project Approach

When we began our project we knew we had three seven week terms to complete it and so we scheduled our goals around that. For the first term we researched effective ways to involve children in Computer Science and drafted a program structure that aimed to keep students involved, motivated, and enthusiastic about the material. During the second term we did additional research and reached out to teachers to get their thoughts on the program. Finally, we utilized the last term to test our program by going into schools to give presentations and doing mock interviews with possible college-level CS students to see what would make the program worthwhile to them.

Part II: Research

Contained within Part 2 is all the research we did to form a foundation for our program. First, we will discuss our research into gender issues and the barriers that women and minorities have to overcome when entering the CS industry. Second, we will include the research we did into the STARS (Students in Technology, Academia, Research and Service) Computing Corps, a program similar to ours which is not yet implemented in New England. Third, we will detail what we learned from meetings with several professionals who had knowledge and experience relevant to our program. Fourth, we will outline the research we did specifically relating to the formation of our program. Fifth, we will overview a number of design decisions we made during our work on the project and our reasoning behind them. Lastly, we outline the research we did for a possible conference at WPI.

Part II A: Gender Issues

1. Catering to Women and Minorities in the Classroom

To try to encourage participation by minority groups and women in the field, we researched different ways to cater to these groups. Future groups working on combating the under representation of women and minorities in Computer Science can supplement their research with the following resources.

In a 2006 article in *Communications of the ACM*, Roli Varma details the barriers to entry facing minority students who wish to study CS in college [15]. Varma's article focuses on underrepresented minority students in a college setting, but provides some general insights into the difficulties they face when pursuing an education in CS at any stage. While we seek to promote CS as a field of study that is open to all demographics, Varma warns that giving special attention to minorities "may convey the subtle message that minorities are not expected to participate fully in CS" (p. 130). When considering how to attract underrepresented minority students to CS, we must take care to be inclusive to them without singling them out.

Varma (2006) notes that students are more comfortable if they have fellow students they can relate to. For students in underrepresented minorities, "The existence of a strong white male culture" (Varma, 2006, p. 132) in CS creates a barrier to entry [15]. Commonly held stereotypes about successful CS students are also typically both intimidating and racially exclusive (Varma, 2006). To assure minority students that they will be accepted by peers in CS, portrayals of CS students should include students from a variety of ethnic backgrounds and highlight non-stereotypical examples of successful CS students.

A 2011 book by the Committee on Underrepresented Groups and the Expansion of the Science and Engineering Workforce Pipeline [CUGESEWP] and the Committee on Science, Engineering and Public Policy Policy and Global Affairs [CSEPPPGA] provides extensive research on minority participation in STEM fields in general [16]. Of particular interest to us are Chapter 6, which addresses some of the social difficulties facing minority students in STEM, and Chapter 8, which includes recommendations for improving minority participation in STEM from various levels of government and education.

The Dot Diva (2010) website provides advice on its “Educators & Parents” pages for making CS appeal specifically to teenage girls [17]. The website stresses the power of images rather than text, detailing guidelines for what kinds of images are most effective. Recommendations include focusing on people creating and using technology rather than focusing on the technology itself, and using photos that “convey the larger meaning” of CS work instead of depicting the work itself. Girls are more responsive to messages that help them understand the impact of CS on the world.

2. Women in IT

As part of our research into the development of our program, we decided to investigate how to engage girls in CS. To do this, we first needed to understand why there are so few girls in CS in the first place, so we read the report, *Girls in IT: The Facts*, which compiles results from several studies to build a more complete picture of the situation [18].

What we found from *Girls in IT* [18] is that there are several factors that can keep students from pursuing computing careers, some of which affect both sexes. Both girls and boys have limited or inaccurate knowledge of what a computing career involves. Computing curricula can often be too theoretical to seem relevant to the real world, which can cause girls and boys alike to lose interest. Also, a classroom decorated with “geeky” imagery can make young women and men who do not consider themselves “geeky” to feel less like they belong. However, there are several other factors that affect women specifically.

Throughout *Girls in IT: The Facts* [18], a running theme that affects women’s interest in computing careers is perception. The report reviews several areas of influence, including family, peers, and media. Perceptions from all of these areas affect girls' interest in CS. For example, the report states that, whether consciously or subconsciously, many adults and peers believe that computing is a masculine field, and that men are naturally better at CS. Even girls will believe this, rating their ability lower than boys with similar levels of achievement. Families are also more likely to encourage their sons to take CS classes. Most girls who take programming classes were encouraged to do so by their family. It should also be worth noting that families that are financially able to provide computers with internet access are more likely to expose their boys to computing, causing these boys to have an advantage when starting programming classes, and therefore causing them to seem more "naturally" talented in computing. This, in turn, can affect teachers' perceptions of student ability. Teachers can have an unconscious bias to believe boys have an "innate" talent for computing, and girls' perception of support from their teachers can have a direct effect on their career interests. Even the media rarely shows women in computing careers, or when they do, they are often portrayed as a "geeky" stereotype which can scare girls away. Changing perceptions about computing will greatly help women feel more comfortable pursuing computing careers, but in the meantime support from teachers, family, and peers can greatly help girls to explore careers in CS.

Part II B: STARS Computing Corps

The STARS Computing Corps is an alliance of many CS departments throughout the US. Among the several programs they run is the STARS Leadership Corps, their main program. The Leadership Corps is similar to our program in many ways; we can learn a lot from it. Like our program, it is a framework that groups can implement. It is designed to encourage students, especially minorities, to participate in computing. Institutions that implement the Corps run it as either a for-credit class or an extracurricular club. The STARS Computing Corps' website provides information that institutions that wish to implement the Leadership Corps will need. There are documents explaining the purpose of the club/course and how to run it. It also provides examples of club constitutions and course curricula from institutions that have implemented the Corps.

Every August, the STARS Computing Corps holds a conference called the STARS Celebration. This is the beginning of the STARS Leadership Corps program. Students from every instance of the Corps are supposed to attend the conference. It acts as a “Call to Action” for new members, and provides a place for existing members to show the work they have been doing. Beyond STARS Celebration, the Leadership Corps program is a wrapper for unique projects. The wrapper consists of “a seminar series addressing the STARS core values, an outreach/service component consisting of Leadership Project teams, a written component (written reflections on service activities), and an oral component (presentations as part of outreach or to disseminate outreach outcomes).” [19] The seminars focus on teamwork, project management, technical skills, communication, and other soft skills. Inside the wrapper, students work on Leadership Projects. This is the majority of what they do in the Corps. These projects involve “outreach to K-12 schools and community organizations, often in partnership with computing professionals who are volunteers.” [19] Most projects are focused on a specialty area, such as game development or robotics.

The STARS Leadership Corps is similar to our project in that it is a framework for college students to help K-12 students get involved in CS. Also like our project, STARS provides a model for institutions to use to implement their own instance of the Leadership Corps. Unlike our project, the STARS Computing Corps acts as a central institution that interacts in some way with every instance of the STARS Leadership Corps. The Computing Corps holds the annual STARS Celebration, which every instance of the Leadership Corps attends. Our project does not have an equivalent central organization. MassCAN serves a similar role for our program in that it will help those who wish to establish their own branch of our program, but it does not provide the kind of detailed framework for doing so that STARS does for its Leadership Corps. Also, the Corps’ Leadership Projects are more varied and unique than what the student volunteers in our project will be doing. The roles of volunteers for our project are more well-defined. They will be running tours, giving presentations in K-12 schools, or helping teachers in their classrooms.

We can learn a lot of useful things for our project by looking at what the STARS Leadership Corps does and how it does it. In general, we could gain useful insights by observing parts of their program that mirror parts of ours. The differences can provide more specific suggestions of things we or institutions running our program might want to change. For example, the Corps’ Leadership Projects are more open-ended than what volunteers for our program do. This might make college students who participate in STARS feel more engaged than volunteers for our project would. Also, STARS has more of a focus on every instance of its program being part of a community. Maybe our project could benefit from something akin to the STARS Celebration. The most useful thing we could do is visit some STARS event, probably the Celebration, to learn more.

Part II C: Meetings

1. Meetings with Mr. Stanton

Jim Stanton, Executive Director of MassCAN, the Massachusetts Computing Attainment Network, was our project sponsor. He helped us establish our goals for the project and divide the project into three phases to correspond with the three terms we would be involved. The first phase was the exploratory phase. This phase was meant to be primarily filled with researching background information for various aspects of our project so we could thoroughly understand the problem we were working with and have all of the information needed to begin working on a solution. An important part of this phase was the development of a survey to collect information pertaining to when and why CS majors became interested in programming. Throughout this phase we also were able to focus the goals of our project.

The second phase of our project - the problem/solution phase - was meant for us to look closely at our problem, and finish designing the potential solutions we had come up with during the first phase. This entailed designing the structure for a college tour to help spur middle- and high-school students' interest in Computer Science as a major. We also decided to develop a program through which college students could work in schools with teachers and young students.

The final phase, or the implementation phase, involved taking the solutions we designed during the second phase, finalizing them, and implementing them to the best of our ability. This last phase materialized into testing our presentation and interview process and then modifying our program using what we learned.

Mr. Stanton also brainstormed with us about a possible conference at WPI, but due to time constraints we decided it was best if we implemented it during the 2014-2015 school year. We did some preliminary research about conferences at WPI to lay the groundwork for a future IQP that would undertake this effort.

2. Meeting with Professor Doyle

During the alpha testing stage of our survey-making process we met with Professor

James Kevin Doyle, head of the Social Science and Policy Studies department at WPI, to discuss survey construction. Most importantly, he recommended that we order the questions chronologically; begin with questions about the survey taker's past and end with questions regarding possible future programs. Professor Doyle also made sure we recognized the difference between becoming interested in a CS major and becoming interested in programming. This was incredibly relevant as our original survey did not specify a difference and as a result misled some of our testers. Professor Doyle helped us understand that it is unlikely anyone decided they wanted to major in CS in middle school, but they may have developed an interest in programming at that age. This helped us refine our survey to be more specific, accurate, and relevant to the target audience.

Additionally, Professor Doyle explained to us that it would likely be useful to include some questions regarding demographics; demographics would help us narrow down what groups we should focus on when working with younger children. By separating survey results by gender we can cater our end results to different sexes.

3. Meeting with Professor Cyr

Once we decided to include a college tour as part of the program, we met with Professor Martha Cyr who has developed a similar tour focused on engineering. This tour, unlike the tour WPI runs for prospective students, is aimed towards middle-school students and does not focus on generating interest in any one field.

The engineering tour includes three main parts. First, there is a short presentation on engineering, focusing on a simple concept important to the field. Then, the students are given an activity relating to the topics discussed in the presentation, such as building a bridge out of toothpicks. Last, the students are taken on a tour of the campus, focusing on where engineering students do their work. If there are a lot of students, they will split into two groups; one group goes on the tour while the other group does the activity, and then the groups switch.

Professor Cyr had a few tips for us for working with middle school students. First, middle-school students like to be entertained, so try to keep things fun. Secondly, information must be kept relevant, not theoretical; Middle school students want to know how things apply to "real life."

4. Meeting with Professor Fisler

During C term we met with Professor Kathi Fisler, a Computer Science professor at WPI whose research interests include computing education, and she had a number of good ideas about where to go with the program. She agreed with Professor Finkel that it was not appropriate to offer students class credit solely for going into the classroom. A class which was taught over the course of the semester and taught pedagogy one term and then facilitated the students going into the classroom the next term, however, had promise. Professor Fisler reasoned that the WPI CS department would see merit in providing such a course, but even if things did not work out, she thought an IQP that worked with students in the classroom was a possibility.

Professor Fisler also expressed interest in either overseeing an IQP working on developing curricula for such a class or overseeing a program like ours. The latter would involve interviewing students to gauge their professionalism and coding skill and putting any suitable candidates in touch with Mr. Stanton.

We also talked briefly about the possibility of having a conference on campus. While she thought it was something that WPI and its STEM program would be very interesting in hosting, she was not sure such a conference would accomplish any more than what CAITE is already doing. CAITE (Commonwealth Alliance for Information Technology Education) is based out of UMASS Amherst and is very focused on increasing opportunities for women and minorities in computing. We decided to ask Mr. Stanton for his thoughts on the program. Refer to http://caite.cs.umass.edu/ for more information about CAITE.

5. Meeting with Ms. Demur

To get a better idea of different interview strategies that work, we interviewed Kimberly Demur who works in Academic Advising at WPI and who is in charge of interviewing and hiring new tutors for the subjects of math and science. While our tutors will not be working with WPI students, the same teamwork skills, flexibility, and mastery of the material that Ms. Demur looks for are very similar to what is needed for our program [13].

Modeling our program after Ms. Demur's, we recommend running two sets of interviews with cuts being made from the candidate pool after each interview. First, candidates should fill out a general questionnaire about their work at college, their hobbies, etc. A document with questions we recommend for this step is included in Appendix B. After receiving filled out questionnaires from applicants, interviewers should look over the GPAs of the candidates. We recommend that any candidates with GPAs lower than 3 should be cut at this point.

The remaining candidates will then be called in for a final individual interview. Before the interview, send them a list of programming problems. These will be used to assess their ability and understanding of coding. While three or four options are provided in Appendix D, applicants should only be expected to complete one of the exercises. During the interview, discuss their thought process to gauge their mastery of the material. Good candidates should be able to accurately communicate their work process, not just complete the work well. We recommend using the document in Appendix C to help record information about their interview.

This interview process should ideally take place the year before the program is run to allow for setup and training. Set aside about four weeks to complete the entire process [13].

It is important to note that Ms. Demur also outlined a group interview process for us to use, but we reasoned that the program was small enough that group interviews would not be needed.

6: Interview with Emily Perlow

We talked to Emily Perlow who has a program similar to ours already implemented at the Flagstreet school. As long as our program does not stray from its original mission of sending tutors into the classroom, students who work for our program can be paid with federal work-study money [12]. This is assuming that students who are chosen for the program have federal work-study and are not already using it for something else. Emily Perlow's contact information can be found in Appendix I.

Part II D: Project Design

1. Survey Construction

In order to design surveys that could return useful data, we first did some background research on survey construction. Our research helped us learn a number of things[3]. We learned that surveys should be kept as short as possible; unnecessary questions should be removed. Questions should be easy for the respondent to answer. The intention of the survey, and any special instructions needed to take it, should be clearly stated at the top. The survey should not ask for personal information unless it is needed. Questions should be consistent and specific. They should not be leading or possibly confusing. Questions should be grouped by subject, and the easiest questions should be at the beginning. Open-ended questions and questions that feel like they have a “right” answer should be saved for later in the survey. The survey should have a clean, organized layout [5][6].

Some information was specific to certain kinds of questions. Multiple-choice questions should be used when the writer(s) of the survey have a good knowledge of the possible answers, and are not looking for new answers. The choices should be exhaustive and mutually exclusive. Questions should be structured consistently. To measure a respondent’s emotion about a subject, a ranking/rating question should be used [8].

There was also information specific to open-response questions. This kind of question should be used when the writer(s) do not know what kind of answer to expect. However, if the writer(s) do have some idea of what answers to expect, the data can be made easier to analyze by offering some prewritten choices in addition to the open-response field. Last, these kinds of questions should be limited in number, because the data they generate is difficult to analyze [9].

2. Tour Activities

We decided to follow the format of the engineering tours that Professor Cyr described, but we needed to replace the engineering activities with some related to CS. We found two categories of alternatives; coding activities that run in a lab and activities that do not require computers.

Scratch [7] is an easy-to-learn, 2D visual programming language created by MIT. It is aimed at middle-school students. Alice [1] is a similar 3D language created by Carnegie Mellon and Electronic Arts. Giving children activities to do with these languages would be engaging and would give them a sense of what programming is like. However, we and other groups implementing the tour would need to reserve enough computers for students on the tour.

Another good option is to introduce students to a robotics program, if available. Robotics provides students with a tangible application of CS that is accessible in the classroom.

The final option would be to incorporate activities from CS Unplugged [2], a collection of CS-related activities licensed under Creative Commons. These activities do not use computers, are easy to run, involve movement, and will lead to more in-depth discussions of CS concepts. Examples of these activities can be found in Appendix E.

Because of the differences between teaching middle-school students and high-school students, the presentations and activities on the tours for the two groups should be different. We have created examples of each that can be found in Appendix E. The presentation in the tour should be related to the activity that the students will be doing as part of the tour. We picked two sample activities for middle- and high-school students, and made a corresponding presentation for middle-school students. We did not make a sample presentation for the high-school tour. A college implementing a tour should feel free to use the examples or make their own presentations and/or pick different activities.

For middle-school students, we recommend the activity “The Intelligent Piece of Paper”, which we found on Computer Science Unplugged. It can be found in Appendix E. We recommend this activity because it shows that computer programs are not “magic”. They are just sets of instructions, and students could make them on their own. This activity only requires a printer to prepare, and pens/pencils and space to run. It is good for middle-school students because it does not require the patience that a computer-based activity would. We have created an example of a presentation that could be shown before running this activity, which can be found as “Example Presentation 2” in Appendix A. The presentation serves as an introduction to the concepts that the activity explores. It also provides some additional structure to the tour, which is important for middle-school students.

For high-school students, we recommend a brief tutorial followed by some free time to experiment with Scratch. This activity requires access to computers. It works well for students in high school because they are more interested in activities with real content. Scratch is simple and intuitive enough that it can be used to do some interesting and creative things in a short amount of time. This activity does not have a corresponding presentation, as the tutorial (which is built into Scratch) is a good introduction to the material. In addition, high-school students are more interested in getting right to the material.

3. Survey Techniques: Past IQP Research

As we developed our survey, we researched past IQP work for any techniques or advice we could use. We found a report written in 2005 by WPI students Laura E. Handler and Patrick A. Hogan about a survey of the career aspirations of Worcester high-school students. The notes and recommendations in the report helped us edit our own survey before initial testing.

The report recommended taking extra care to make the most crucial questions clear and visible to survey takers. The authors of the report had written a survey where demographic information was key to their research. Thus, they had to discard any data from surveys where the demographic questions were left unanswered [4].

4. Interview Techniques

A good interview requires preparation in order to ensure that the right questions are asked. In our case, we need to interview students to find good mentors who we will train to work in the classroom with middle-school and high-school students. When interviewing potential mentors it is important to look for a number of traits.

When interviewing, we reasoned that there is an ideal mentor we are looking for. As expressed above, they need to be a CS major because they need to have a wide breadth of experience with the material being taught in schools. In the same vein, a mentor selected for the program should be finishing their junior year; this way they will have in-depth experience with programming but will also have the opportunity to attend summer training.

Mentors should be able to speak fluent English so that they can accurately understand instructions and get their points across. It is important that those working with the mentor can understand the help they are receiving. If the school teaches in a language other than English, it is important for the tutor to fully understand that language.

It is also important for mentors to be comfortable working closely with others; mentoring is often a different process than teaching. Rather than talk at the student or teacher, mentors need to be able to pick up on visual cues and adjust their teaching style if they see that the teaching method is not working. With this in mind, they should also be able to adapt to different ways of learning and teach in the way that works best for the student or teacher they are working with.

Try to look for mentors who are patient, as CS can be a complicated and confusing topic to someone who has never worked with code before. It is even better if they have experience working in the classroom with middle-school or high-school students.

For our mentors to be most effective in the classrooms, they should be punctual, focused on their work, productive, courteous, honest, and eager to work cooperatively with others. In this way they can help contribute to a supportive, positive learning environment. The grades of prospective mentors are a good measure of these traits. See the interview process below for more information on what to ask candidates and how to assess their eligibility for the program.

5. Background Checks

Since students working as a part of this program will be working alongside teachers, we anticipate that they will not need to request a CORI check. However, as the nature of this program may change in the future, information about requesting a CORI check has been included below.

Massachusetts law mandates that a student working with other students without supervision must take a CORI check at least once every three years. Since the form can take some time to process, it is recommended to request one immediately after the interview process is completed. Different jobs require different CORI forms. Working in schools will require two. To view which forms are required, refer to the CORI Reform Law, located in Appendix F, for a sample of a filled out CORI request form as well as a link to the appropriate site. To request a form of your own, see Appendix G.

6. Financial Aid and Funding

For students in college, time is a very limited resource. Every student has a multitude of things which vie for their time including schoolwork, homework, sports, sleep, and jobs. If we are asking the students to add another commitment to their already packed schedule there should be compensation involved. This way, the student will not feel pressure to work with our program as well as hold down another job.

Furthermore, we are expecting a large volume of work from them; an after-school commitment would probably work best 3 times a week. In addition, they will be setting up and presenting an introduction to computing and, later in the program, setting up and orchestrating a tour for the students. Any one of these tasks would be reasonable for a volunteer, but together they compose a significant workload.

If a student is getting paid for their work they will make it a priority. Students will make the extra effort to proofread their work and make sure it is of the highest quality. Volunteering, on the other hand, is something students may neglect until everything else is complete.

In terms of resumes, volunteer work and paid work are also seen differently by employers; a paid job is of higher worth than a volunteer position. If students are paid, the program will have the added benefit of attracting those looking for an internship. Such work also serves as valuable publicity for MassCAN.

It could be argued that providing funding for the program will attract people looking to be paid, not people interested in helping students learn about the Computer Science industry. That is why interviews are a necessity; they will separate those who are legitimately interested from those just looking for money.

Without funding, this program can still proceed. Keep in mind, we have secured 15 hours of pay from federal Financial Aid. See Appendix H for different work schedules which take into account different funding levels. The unfunded, or volunteer option, does not involve going into the classroom; it is missing an integral part of the program. We highly recommend setting aside funding for the program so that all aspects can be implemented.

Part II E: Design Decisions

1. Classroom Involvement: 8th Grade vs. 10th Grade

Mr. Stanton identified middle-school students as the most critical age group for us to target, but clarified that high-school students are also important to focus on. We reasoned that eighth- and tenth-grade students are good target groups but in the end it is important to remember that college students will be working with whichever students are in the participating school's Computer Science course.

Regardless, middle-school and high-school students are different enough that the presentation, tour, and especially classroom involvement should be at least somewhat different for each group. To figure out what those differences should be, we researched differences in teaching eighth- and tenth-graders. Adam also had some experience working with the two age groups from his job as a computer camp counselor.

According to our research [10], middle-school students are just starting to explore their independence, and are harder to get involved in adult-led activities. Middle-school students tend to disengage from school and structured activities. This is strongest (unfortunately for us) in eighth grade. Eighth-grade students will have a strong desire to be with their friends, and will be much less interested in an activity if they are forced to separate. However, they will have a stronger interest in an activity if they can do it with their friends. They will want to do something if their friends are doing it.

While it may seem counterintuitive given the last point, middle-school students need a structure in place to help them feel safe and give them a way to grow. There should be room to explore within the structure, but overall it should be consistent. Routines are especially important.

Middle school is a sort of “tipping point” where students will either more strongly cement their interest in something or start to close themselves off to it. This is why middle school is the most critical age for us to target.

High-school students are more independent. They will want to decide for themselves how to spend their time. They will be more focused on a small number of areas of interest. High-school students know what they want to learn and focus on, so they will be much more interested in activities that focus on what they want to learn and much less interested in activities that do not. High-school students are more interested in content than students in middle school.

Older students should be given more responsibility. This means giving them more freedom to complete tasks the way they want to, but also means being less willing to accept excuses if they do not complete something. High-school students are often concerned about college. Demonstrating how their work applies to college will help students appreciate it.

2. Classroom Involvement: Number of Mentors

In our meetings the idea of possibly hiring multiple mentors came up, and we decided to pursue it to see if the idea had merit. It was important to consider whether students should work individually or as a team. By themselves, students would have an easier time scheduling and working on the project, but might be overwhelmed with the work load. Two students might have a bit more trouble scheduling meetings and agreeing on times of the week to get involved after school, but they would be able to split the work of designing and presenting the presentation as well as leading students on a tour.

3. Design Decision: After School or in Class

During our design process we had to decide whether working with students after school would be more effective than working during class hours. There were fundamental differences between middle-school and high-school classes that we took into account.

A middle-school curriculum is generally less defined than that of a high-school classroom; it would be unusual for a student to be required or even encouraged to stay after class for additional learning. With this in mind, we decided it is best to have middle-school programs be involved directly in the class day. If the teacher requests help after school, it is most likely that only one student will be necessary.

Even in the high-school classroom there is no guarantee that students will take advantage of tutors working after school. Once again, if the teacher requests after school help there are a number of things the teacher can do. For one, the teacher could incorporate staying after school as part of the curriculum; this way, students would be required to participate. We do not recommend requiring the students to come after during every visit to the school; both high-school and middle-school students have other commitments like sports or homework. Instead, ask that students stay after to work with the assistants three or four times over the course of the semester. Hopefully, the assistants' help will be valuable enough to them that they return for more.

All things considered, it is definitely most beneficial to have college-level mentors work during the school day as it will make the best use of students' time. After-school work should only be undertaken in special circumstances.

Part II F: Conferences

As the program came together, we decided it was time to look for feedback regarding how effective our program would be when applied in the classroom. Mr. Stanton suggested a conference would be an ideal way to do this. However, we evaluated the time we had left and, realizing that teachers need significant advance notice for day-long commitments, we decided instead to lay the groundwork for another IQP that would continue our work next year.

The centerpiece of the conference was having a few prominent keynote speakers who could foster conversation about the industry in general and what strategies had and had not worked to get students more involved in CS. Mr. Stanton suggested we invite a keynote speaker from Google as well as asking Professor Fisler to speak.

However, before we could get in touch with any speakers for the conference we needed to look into a number of facets of a successful conference in advance. Most importantly, we needed a space to meet, and Professor Finkel also advised that we research different food options, research how to best reserve parking, and ask possible guests which day of the week works best for them.

In terms of food, we decided that overlapping the lunch hour was best; this would determine to a large degree what food we would serve. WPI has a catering service that anyone holding an event on campus needs to order from. Prices range from about 8 dollars per person for a continental breakfast to 25 dollars per person for a full lunch.

It is important to remember to reserve parking because visitors often need to have a parking pass to avoid getting fined; be sure to include instructions for obtaining such a pass and detailed directions to get to campus in any invitations that are sent out. We called the campus police, who were nice enough to guide us through the process. When setting up a conference, the campus police are often in charge of parking and talking to them can ensure that the process goes smoothly.

To make sure that there is space available, make sure to talk to the events office and set a date months before the conference takes place. Also be sure to reserve an area large enough to seat the audience Mr. Stanton anticipates: close to 75 people. Rooms at WPI that seat that many people include Upper Fuller Auditorium and Olin Hall 113.

It is also probably a good idea to record the conference. We recommend either borrowing cameras from the ATC or using a room which has lecture capture, which is used to record class lectures. Since we were planning on having a panel of speakers, having a number of cameras seemed more likely to be successful. It is a good idea to have someone from Lens & Lights, the organization on campus which deals with lights and sound, come and help run the conference so that each speaker can be properly equipped with a microphone.

Finally, make sure to send out invitations in advance. Mr. Stanton recommends 6-8 weeks, but the earlier the better. Mr. Stanton suggested inviting teachers from all the colleges in Massachusetts which have a CS major along with some of their students.

PART III: Survey Examination

Part III A: Survey Results

To try to improve our program, we surveyed CS-related majors, Computer Science and the Interactive Media and Game Development Tech track, to figure out what made the students get involved in the field. Over the course of a week we got 90 responses, which we attributed to a number of publicity measures we put in place; we arranged for an Amazon gift certificate to be raffled off to one of the survey-takers, and mentioned this raffle in all of our emails to the student body. We also printed out fliers with QR codes that linked directly to the survey and left the fliers in computer labs where CS majors work.

If we had not had such a good response, we had a number of options lined up including printing flyers or standing outside of CS classrooms and handing out the link to the survey as people left class. It is important to focus on areas where our intended audience frequently gathers; other options include CS labs and lounges.

To start off the survey, the questions and results of which can be found in Appendix J, we asked when the respondents first became interested in programming. As we predicted, the highest percentage of students became interested in high school; 42 percent became interested in high school, 34 percent in middle school, and the rest either became interested in elementary school or in college. This statistic enforces the importance of having both a middle-school and a high-school program; both will help cement a student's interest in the field.

We also asked a slightly different question of these people who were interested in programming: when did they decide to pursue a career in the industry? We discovered 67 percent of respondents made this decision in high school. With this in mind, it is important to focus on the interesting aspects, as well as applications, of CS during the high-school college tour so that students know what the field has to offer.

When asked what made them decide to be a programming major, sixteen percent of respondents mentioned an interest in upcoming technologies and a desire to change the world. By incorporating upcoming technology into our presentations and tours, we can help stimulate this interest. Another 50 percent became involved because of a love for coding. Unfortunately, this is primarily an internal motivator or controlled by the class's curriculum, but we recommend encouraging students to pursue further research outside the classroom regardless because it will give them the opportunity to work on material and projects that they enjoy rather than work they are required to complete.

We compared the answers to the questions of when students decided to major in the subject and whether they had had any programming courses in high school. From this we noticed that roughly 80 percent of students that decided to major in CS after already attending college for some time had not had any programming courses in high school.

Another correlation was found between the happiness of CS majors and whether or not they had taken any programming courses in high school. Almost all students that rated their happiness on the lower half of the scale had not taken any programming courses in high school, nor studied the topic on their own.

Lastly, we needed to gauge interest in working in the pre-college classroom; if no one applies for a mentorship position, the program will not be viable. Of the 90 responses, 26 were interested; assuming this can be extrapolated to other colleges that have CS classes we have more than enough interest to move forward.

The survey revealed a number of other statistics, but they were not directly relevant to making our program successful. They may, however, be useful for future research involving college students working in the high-school classroom. The full results of the survey can be found in Appendix J.

Part III B: Recommendations for Future Surveys

The time gap for the audience was one of the biggest problems our survey encountered; most people who took the survey reported having trouble remembering why they became interested in the field. On a similar note, most respondents noted that it was hard to gauge whether a campus tour would have been helpful in guiding their career decision.

If another, similar survey is completed in the future we would also recommend asking a few questions that we only realized the value of after we had received the results of our survey. Most notably, it should ask CS majors who changed their major or decided to major in CS in college what their major was beforehand, if any.

PART IV: Testing

To ensure that what we had developed would be successful, we visited the Advanced Math and Science Academy (AMSA) to test a presentation we had created to introduce the program. We gave our initial presentation to a mixture of middle-school and high-school students. Results of the testing can be found below.

Part IV A: Sample Presentation

We presented Presentation 1, included with Appendix A, to a group of about 40 middle- and high-school students. The initial presentation took about 10 minutes, and then we took questions which filled the half-hour we had available. We did not need to prompt the audience for questions.

We had a questionnaire to go with the question-and-answer session, but the audience asked so many questions that we did not have time to administer it. Instead, we asked for feedback by show of hands on a few of the important questions. Students who watched the presentation and question-and-answer session were more interested at the end of the program than at the beginning, and also were eager to have college students work alongside them in the classroom.

When asked, the audience indicated that they would have preferred to have more time allotted for asking questions rather than viewing the presentation. To deal with the shortage of time allotted to answer questions, we decided to shrink our presentation and keep the presenter notes on hand to answer questions and start conversation; students asked questions regarding similar information.

Second, we added information about our program to the slides. This helps clue students in to the fact that we will be available in the classroom for them. We also noticed that some, though few, of the students had trouble staying interested. To help remedy this, we added concrete examples of what different students at WPI have done with programming. These examples are also helpful when answering questions.

Part IV B: Sample Interviews

We held interviews on WPI campus, and while there was widespread interest, none of the students we talked to were able to make the time commitment necessary to participate in the program next year. When asked to recommend something that would make the commitment more valuable to them, a huge majority of interviewed students requested class credit. Lesser obstacles included transportation, as a number of students were interested but did not have a way to get to the school each week. If you can, find a pair of students per school so that they can carpool, or send students to schools that are within walking distance. Closer schools also mean less travel time, which is always an issue for busy college students.

To make this program most successful, class credit should be secured as an option for students. Rather than simply offering credit for working in the classroom, however, see if your CS department is interested in offering a course on the pedagogy of CS which includes a practical component. Be sure to provide transportation for students if it is required for them to work in the classroom. If a class were to take form, its designers should contact the WPI IGSD and inquire as to what methods the school uses to facilitate off-campus transportation. Other schools implementing the program should look to any campus organization that offers frequent transportation for a significant number of students such as sports teams.

Part V: Future Research and Programming

Part V A: Holding a Conference

Several things need to be handled while preparing to throw a conference. In addition to coming up with a schedule for what is to occur during the conference there are a few logistical matters that need to be addressed. Hosting a conference for a large number of people is no small task. There are three major aspects of hosting a conference that need to be addressed beforehand: parking for guests, refreshments, and space to hold the event.

Procuring parking spaces for the conference guests was, in our case, as simple as contacting the campus police. Providing the campus police with preferred dates and times, location on campus, and number of people attending will make it easier for them to provide you with parking. This should be done before informing your guests of a date in case parking can only be provided on certain days due to other events.

Secondly, the dining services on campus may be able to provide catering for the conference guests. The price varies with the number of servings that are prepared so an estimated head count is recommended when contacting the campus dining services. If this option does not work, look into hiring a third-party catering service.

The method for reserving a space to hold the conference varies widely by college. You will have to look into how to reserve rooms at your institution. Remember to keep the expected number of attendees in mind and reserve space well ahead of time.

Part V B: Closing thoughts

Looking back on all the research we did for this program, we think the framework we have put together will serve as a valuable starting point for any future programs that are implemented in Massachusetts schools.

Interestingly enough, the end goal of the program has changed since its conception. Instead of a standalone program, we recommend incorporating our three-step program into a class which teaches students pedagogy during the first half and then coordinates the students' involvement in high schools and middle schools. This resolves a number of roadblocks we hit while designing the program including mentor motivation, transportation, and finding a good way to prepare students for their work.

References

1. Alice. (2013). Retrieved 10/9, 2013, from http://www.alice.org/index.php
2. Computer science unplugged. Retrieved 10/9, 2013, from http://csunplugged.com/
3. Guilford Technical Community College. Survey question construction techniques and tips. Retrieved 9/12, 2013, from http://www.gtcc.edu/media/12359/surveyconstructiontech.pdf
4. Hogan,Patrick A.Student author -- ME, Handler,Laura E.Student author -- ID, & Wilkes,John M.Faculty advisor -- SS. (2005). Gender-based comparative survey of public HS students(Undergraduate Interactive Qualifying Project Number 05C022I). Retrieved from Worcester Polytechnic Institute Scanned Projects Collection: http://www.wpi.edu/Pubs/E-project/Scanned/05C022I.pdfhttp://www.wpi.edu/Pubs/E-project/Scanned/05C022I.pdfhttp://www.wpi.edu/Pubs/E-project/Scanned/05C022I.pdf
5. Lime Survey. (2013). How to design a good survey (guide). Retrieved 9/17, 2013, from http://manual.limesurvey.org/wiki/How_to_design_a_good_survey_(guide)
6. Science Buddies. (2013). Designing a survey. Retrieved 9/17, 2103, from http://www.sciencebuddies.org/science-fair-projects/project_ideas/Soc_survey.shtml
7. Scratch. Retrieved 10/9, 2013, from http://scratch.mit.edu/
8. Seibert, T. (2002). Designing surveys that count. Retrieved 9/17, 2013, from http://www.keene.edu/crc/forms/designingsurveysthatcount.pdf
9. The College of New Jersey. How to make a good survey. Retrieved 9/17, 2013, from http://assess.pages.tcnj.edu/?page_id=177
10. The Wallace Foundation. (2013). Developmental differences between middle school and high school programs. Retrieved 10/9, 2013, from http://www.wallacefoundation.org/pages/chapter-3-engaging-older-youth.aspxhttp://www.wallacefoundation.org/pages/chapter-3-engaging-older-youth.aspxhttp://www.wallacefoundation.org/pages/chapter-3-engaging-older-youth.aspx
11. Meeting with Kim Demur, Academic Advisor and person in charge of interviewing Tutors at WPI
12. Email communication with Emily Perlow (10/13)
13. Meeting with Kimberly Demur, Academic Advisor at WPI (11/5)

14. Peer Academic Coach Training, WPI 2012-2014
15. Varma, R. (2006). Making Computer Science minority-friendly. *Communications of the ACM. 49* (2), 129-134)
16. Committee on Underrepresented Groups and the Expansion of the Science and Engineering Workforce Pipeline (U.S.), Committee on Science, Engineering, and Public Policy Policy and Global Affairs (2011). *Expanding underrepresented minority participation: America's science and technology talent at the crossroads*. Washington, DC: National Academies Press
17. Dot Diva (n.d.). Dot Diva | Educators & Parents | What Works with Girls: Messages. Retrieved November 9, 2013 from http://dotdiva.org/educators/messages.htmlhttp://dotdiva.org/educators/messages.htmlhttp://dotdiva.org/educators/messages.html
18. NCWIT, Ashcraft, C., Eger, E., Friend, M. (2013). Girls in IT: The facts. Retrieved February 23, 2014 from https://www.ncwit.org/thefactsgirls
19. STARS Computing Corps (n.d.). Corps Process and Components | STARS Computing Corps. Retrieved March 5, 2014 from http://unccslc.starsalliance.org/corps-process-and-components

Appendices

Appendix A: Program Start Up Guide

Appendix B: Initial Interview Questionnaire

Appendix C: Individual Interview

Appendix D: Interview Questions to Gauge Knowledge

Appendix E: Tour Activities

Appendix F: Cori Checks

Appendix G: Cori Check Source

Appendix H: Work Schedules and Funding Levels

Appendix I: Contact Information

Appendix J: Survey Results