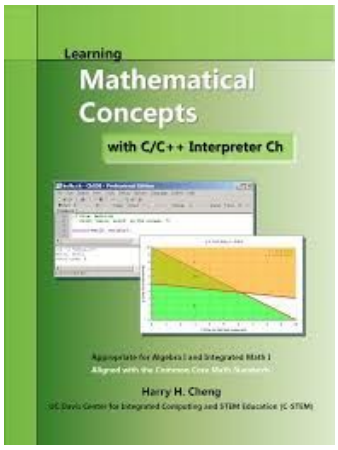
Clay Dagler's Portfolio

CREST (Computing Research Experiences for STEM Teachers) **Total Hours: 400**
Dates: 07/02/2012 - 08/10/2012, 07/08/2013 - 08/02/2013

I led trainings for teachers on how to use computing and robotics to teach mathematics over the first two weeks of the summers listed above. A majority of these teachers had no programing experience. The first week we focused on intro to programming and how to use it in a math classroom. The topics in this week include variables and their types, input/output, debugging, if-else statements, loops, and plotting. In the second week I led trainings on how to use robotics to help students learn mathematics. The topics covered in this week included what a robot is, joints and their angles, the zero position, relative vs. absolute position, blocking vs. non-blocking, sensors and modular robotics. All of the robots were controlled by writing code which required understanding the topics covered in the week before.



For the remaining days of this research experience I worked on writing group work activities and chapters for different books for the C-STEM center. These activities included writing computer programming, documentation in LaTex, and several of them included video lessons for the unit. Topics included in these chapters and activities include using loops to generate tables to explore linear, quadratic and exponential functions; using functions in programming to understand functions in math; basic numerical analysis; if-else statements to understand absolute value and piecewise functions; and writing code to generate animations. Some of the activities I focused on included topics beyond high school Algebra; for example, I created several animations that shows tangent lines to different functions.

YCOE CaMSP C-STEM+ Total Hours: 150

Dates: 07/14/2014 - 08/01/2014, 07/13/2015 - 07/23/2015, 07/18/2016 - 07/28/2016

This is a three year grant where the teachers get a total of 60 hours of instruction and 72 hours of follow up meetings. For this grant I taught programming topics to a room of 85 plus teachers, and I was in charge of supporting a small group of 14 teachers from four different districts. My first goal in this group was to make sure each teacher had enough computer programming knowledge to cover it in their classes. Once we achieved this goal, for the remainder of the time I supported them in developing high school math activities which incorporate coding to use in their classrooms. During the school year we met quarterly to discusses how the programming lessons are going and I provided additional support for the teachers. Even though the teachers only meet 132 hours for this grant, I work an extra 28 because I was supporting other small groups.

1,2 and 4 Day Trainings Total Hours: 176

Dates and Locations

06/24/2013 - 06/28/2013	@ UC Davis	04/21/2014	@ Hayward
07/01/2013 - 07/05/2013	@ UC Davis	06/23/2014 - 06/26/2014	@ Hayward

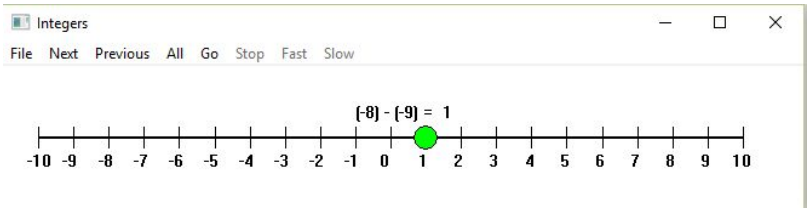
11/03/2013	@ UC Davis	11/27/2014 - 11/28/2014	@ Los Angeles
01/09/2014 - 01/10/2014	@ Riverside	02/26/2014 - 02/27/2014	@ UC Davis
01/11/2014 - 01/12/2014	@ Orange County		

These trainings are very similar to the two week trainings that I led in the CREST and YCOE grant explained above. I taught the attending teachers how to use computer programming and robotics in their math and science classrooms. The one day trainings were an introduction to the UC Davis C-STEM program and how computer programming can help students learn mathematics and science. The teachers that completed this two day training gained enough experienced and material to incorporate coding and robotics into their classrooms. The four day trainings had more time built in for the teachers to engage in hands on and cooperative learning.

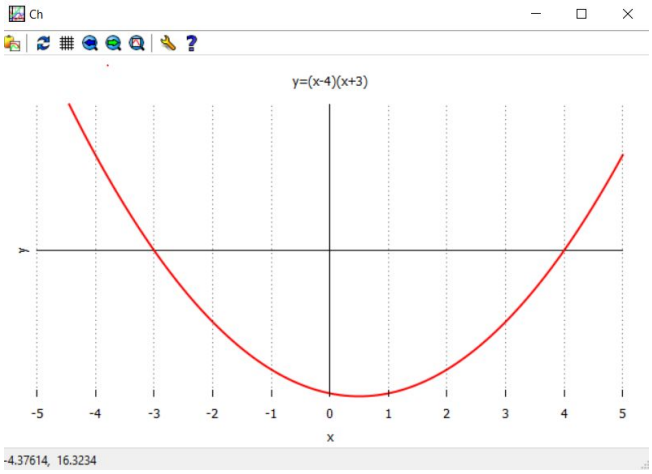
The UC Davis C-STEM center also hired me to do a one-on-one training on March 15th and April 19th, 2015. The teacher I worked with during this time learned the fundamentals of computer programing and got ideas how to use it in her classroom.

Created Coding Activities For the UCD C-STEM Forum — Total Hours: 420

Dates: 06/2012 to Present



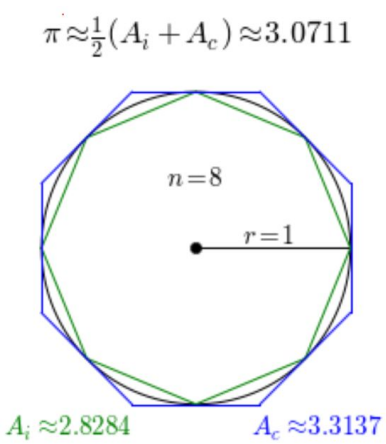
The UC Davis C-STEM center has a forum where teachers share good activities that use programming and robotics for the 6th through 14th grade classroom. Dr. Cheng has asked me to create and upload computer programming activities to share with other teachers that are part of the UC Davis C-STEM program. One of the most important pre-skills in high school math is the ability to add and subtract integers. To help the students master this, I created a computer program that generates random animations which shows how to add and subtract integers on a horizontal number line. Order of operations is also a skill students need in high school mathematics. I created a programming activity that has the students use only four sevens and a combination of mathematical operations to have their result be as many numbers as they can between 1 and 100. This activity also introduces integer division and modular arithmetic in the C programming language; which is a great way to review order of operations because it takes it to a more advanced level which the students have never seen before.



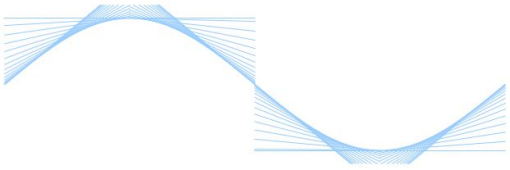
Even though students have seen equation and inequalities before, it is still a big topic in a freshman math class. To help with this topic I created activities where students first solve equations and inequalities to check their work by writing a small computer program. I also created a computer program that helps students understand positive and negative slope as well as y-intercept for a line. This program generates a controlled random graph of a line with no grid or tics shown. Once the graph is outputted, the program displays four different lines in slope intercept form with one of them being the line which matches the

graph. The task for the student is to find the equation that matches the graph. Once the students know the graphs of $y=|x|$ and $y=x^2$ they need to be able to graph $y=a|x-h|+k$ and $y=a(x-h)^2+k$. In order for the students to discover how to graph these equations I created a set of programs where the students input the values for *a*, *h*, and *k* and the program outputs the graph of the absolute value or quadratic function. Once the students discover how *a*, *h,* and *k* affect the graph, they will be able to sketch these graphs by hand and write down the equation when given one of these graphs.

Factoring quadratic expressions is a big topic in Algebra 1. Once the students have a good understanding of this, it is time for them to see how a quadratic in factored form connects to the roots of its graph. To help students with this I created a program that graphs a quadratic equation with its roots clearly labeled. The students use this program to discover the relationship between the roots of the graph of a parabola to its equation. Another concept the students are introduced to at the end of Algebra 1 and master in Algebra 2 is finding the vertex of a parabola. Once the students know the formula for this they use a program I created to help them master this skill. The students first find the vertex of a parabola in standard form ($y=ax^2+bx+c$) and then check their work by running the program I created. This program reads in the values for *a*, *b,* and *c* and then generates the graph of the quadratic. The students then use their understanding of quadratic equations and the output from my program to check their work. To introduce students to solve 2D systems of linear equations I created a program which has two simulated robots on a *xy-axis* drive straight from different starting prints and the students are faced with finding where the two robots meet. Mathematically, this is the same as solving algebraic systems. A program I uploaded that is covered in Geometry involves calculating the area of different regular polygons. To enrich this standard I wrote a computer program that creates an animation that calculates the area of several polygons about a circle of radius 1 and shows the results with a picture. This shows how the area approaches π as the numbers of sides of increases. There is another program that goes with this activity that generates these calculations in a table in HTML format.



Activities that I have uploaded to the C-STEM forum that extend past algebra 2 are covered in Statistics, Pre-Calculus and Calculus. For Statistics I wrote two programs. The first one creates a table used for the calculation of the standard deviation for a set of data. Another program I wrote uses random numbers to simulate the sum of two dice. The computer "rolls" these two dice many times and keeps track of the number of times their sum is 2, 3, 4, …, 12. Once all of the calculations are down, the program outputs a histogram of the results to show a bell curve. Finding tangent lines to a given curve is one of the top two topics in first year Calculus. I wrote and uploaded several computer programs that generate animations which show tangent lines to several curves. Calculus was used to find the equations of these tangent lines and they were graphed along the curve at every .1 interval. The other large topic in first year Calculus is finding the area under the curve by using several rectangles with ridiculously small widths. To help students understand this topic I created some code that generates a graph to show this for the equation $y=0.25x^3$. This program reads in from the user if he/she wants the lower or upper bound for this approximation and the width of each rectangle. Then the program outputs a graph showing the function and all of the rectangles, the total area of all the rectangles and the actual area under the curve. These visuals give the students a strong understanding of integration.

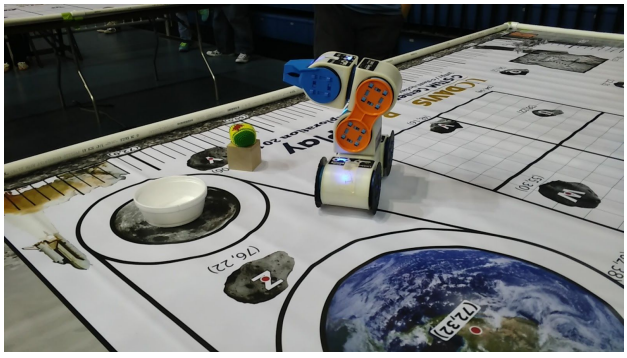


More and more California schools are going away from the Algebra 1, Geometry, Algebra 2 track and replacing it with Integrated Math 1, 2 and 3. This means students are now seeing topics in their freshman and sophomore classes that they have not seen before. Some of these topics include scatter plots, box and whisker plots, and bar graphs. For the scatter plot standard I created an activity where all the students first measure their height, wing-span and face length. Then the students use the data from the class to write two programs that create scatter plots, one for height vs. wing-span and the other for height vs. face length. I wrote a program that generates box and whisker plots for a given set of data. The first thing the students do in this activity is to find the the max, min, median and quartiles for the data. Then the students make small modifications to my code to generate a box and whisker plot for their data. The students then use their work and the outputted graph to discover what a box and whisker plot is and how to draw them by hand. I also created a computer program that generates bar graphs for a set of data and the graph's class interval. The students modify the program by not changing the data but the size of the class interval to see how it affects the graph. In fact, other programmers for the C-STEM center used my code to create a program that generates histograms.

It is not only the topics covered in Integrated Math that differs from the traditional track but also how it approaches them. Applications show up more often in the Integrate Math classes. Another concept that shows up much more is connecting graphs, tables and equations. To help my students master these standards I created several activities that align to these standards and include programming. An example of this is the students face several word problems that they have to first write an equation that models the situation and then modified some code I wrote to generate a table in order to find the solution to the application.

C-STEM Day (Roboplay Competition) Total Hours: 400

Dates: 05/05/2013, 05/31/2014, 05/30/2015, 05/21/2016



The RoboPlay Challenge Competition is a theme-based level playing field robotics competition for K-14 students. It is designed for students to showcase their real-world problem solving skills in a competitive environment. This competition simulates an unexpected problem occurring at a remote location such as a space station or planetary habitat, where a robotic solution must be quickly developed and deployed, using only existing resources. The competition challenges students to creatively use modular robots and accessories to complete various tasks. The competition arena and specific challenge will be unknown to participants until the day of the competition. Using their math, programming, and problem solving skills, students try to most efficiently obtain the highest score for each task on their own.

In an after school class my students learn the robotics and programming skills they need to be successful in this competition. Each year I do this I have to create new activities that include programming and robots for my students to work on. The coding skills in these projects include: parsing a string, loops, if-else statements, input/output, creating graphs and functions. The computation problems can be found at: http://c-stem.ucdavis.edu/c-stem-day/overview/c-stem-day-resources/. Another part of this competition is the RoboPlay Video Competition. For this part of the competition my students create programs that control the robots to make a small skit. To help my students with this I have created several programming activities which cover the necessary programming skills to make the videos. Once the

programming is done, the students make props for the skits, recorded the robots "acting" and edit the videos. The final product is then uploaded to be judged by the volunteers of the C-STEM center. Here are some links to my students' past videos:

- World War II (https://www.youtube.com/watch?v=IXBJx7WVXT8)
- The Walking Robot (https://www.youtube.com/watch?v=JNrPgcEOdRc)
- The Hungery Caaterpillar (https://www.youtube.com/watch?v=qNFg66SNwps)
- Robot 101: How to make a video for the RoboPlay Video Competition (https://goo.gl/QZ5su6)
- Monster Hunter (https://www.youtube.com/watch?v=O6sFxxXts44)
- Breaking News (https://youtu.be/z_grHU1Qua8)
- Girlbot & Boybot (https://youtu.be/sfaQ1v61SO8)

CoodForHood Event — Total Hours: 20

Dates: 09/11/2015 - 09/13/2015

At this event, students from 12 to 18 years old learn coding, video game design, 3D design, engineering, art, website design, and entrepreneurship. I took a team of five students. For the first two days, my students chose different workshops to attend. On the third day my team of students competed by creating a light show using a large circuit board and presented their project to a panel of judges made up of local programmers.

Code Wars — Total Hours: 20

Date: 03/05/2015

This is a programming competition for high school students that is hosted across the world at different Hewlett Packard facilities. I brought a team of three students to compete at this computation. These three students meet with me several times after school before this computation where we went over the programming problems from past computations. HP took very good care of us at this event: all of the problems where very challenging, great food, a raffle with expansive prizes and a generous thank you package.

C-STEM Conference — Total Hours: 30

Dates: 05/18/2013, 05/31/2014, 11/07/2015

This is a yearly conference that focuses on computing, science, technology, engineering and mathematics education. At these conferences I presented at one or more sessions. In 2013 the title of my talk was "Integrating Computing and Robotics into Algebra I with ELL Students — Group Teaching and Collaborative Learning"; in 2014 the title of my talk was "Gaining Understanding of Mathematics Using C-STEM"; and the title of my 2015 talk was " Integrated Math 1&2: Starting an Integrated Approach in High School Math". The overall theme for all of these talks was how I use computer programing to enrich and improve my direct teaching and activities I do in my math classes.

Emerging Technologies 2016 - Smarter Government Through Innovation — Total Hours: 20

Date: 05/12/2015

This conference is put on by the California Department of Technology. The conference focuses on: Security and Risk Management; Emerging Government solutions like Cloud Services, Mobile technologies, & Customer Relationship Management Tools; New Methods of development like Agile and Incremental Software Delivery; Data Management including Open Data and Open Source, Business Intelligence and Data Analytics; Operational Challenges like Consolidation and Optimization, Disaster Recovery, Business Continuity and Legacy Modernization; and Business Issues like, Budget and Cost Control, Human Resources/Talent Management, and the Development of an Enterprise Vision and Roadmap for IT. For this conference I had a demonstration table where five of my students and I showed individuals that work in technology how I teach computer programing and robotics throughout all of my classes. I also gave a ten minute talk at lunch on how I am preparing my students to work in technology fields like computer programming and robotics. The link to this talk can be found at: https://goo.gl/sPqOj9. At this conference I was awarded the Innovative Educator Award.

Extra C-STEM Summer Work — Total Hours: 40

Dates: 07/07/2014 - 07/25/2014, 06/29/2015 - 07/10/2015

During these dates I wrote new chapters and added to existing chapters for the textbook: *Learning Common Core Mathematics with C/C++ Interpreter Ch*. These chapters include everything you would find in a typical math textbook: definitions, example problems, and problems for the students to practice. How this textbook differs from others is that it includes computer programming to enrich the learning experience. The chapters I wrote were: Operations on Fractions, Solving Simple Algebraic Equations, Solving Multiple Step Algebraic Equations, Multiplying Polynomials, Factoring Polynomials, Functions, and Radical Expressions and Equations. The rest of my time during this work was spent editing, adding topics to existing chapters and restructuring the book.

Creating and Editing Course of Studies — Total Hours: 120

Dates: 07/01/2012 - 07/01/2016

For the past four year I have been changing the way students learn math by adding computer programming and robotics to improve instruction. I have discovered that it is a great deal of work when you are on the forefront of something like this. New courses have to be created when you are teaching content a new way, which means that I needed to write new course of studies for my district. The first course of study I wrote for Sacramento City Unified School District was Learning Algebra Using C/C++ Interpreter Ch in 2012. This is an elective course for freshmen that struggle in mathematics and need additional help. The first year I taught this was to a two hour SDAIE math class. For the first of these two hours I used the computer programming and robotics to support my students understanding. In the second hour of this two hour block I taught my students in a traditional classroom setting.

Over the next few years I taught this math class with computer programing as an elective class for students that needed the support in their traditional math class. Unlike the SDAIE class I taught, I only had these students for one hour and they had a different teacher for their traditional math class. For these years the other teachers and I worked closely together to make sure we were meeting the needs of our students.

A vast majority of the high schools in the state of California are not able to have their freshman in two hours of math because of staffing and other graduation requirements. To solve this problem Dr.

Cheng hired a team of 10 teachers to write two A-G approved course of studies in a short period of time. I was on this team and we worked for three days from May 16th - 18th, 2014. The titles for the course of studies we worked on are: *Algebra 1 with Computing* and *Algebra 1 with Computing and Robotics*. These course of studies differ from what I taught in the past because these classes give the students math credit instead of just elective credit, which means that they only need to take one math class their freshman year.

When my district switched from Algebra 1 to Integrated Math, Dr. Cheng asked me to write a course of study for my district called *Integrated Math 1 with Computing & Robotics (CSTEM)*. This course of study is very similar to the ones I wrote with a team of teachers because it give the student math credit so there is no need for the students to be in two hours of math their freshman year. I first submitted this course of study September of 2014 but it was denied by my district because they did not believe that all of the content in Integrated Math 1 could be covered at the same time the students were learning programming and robotics. To solve this problem, the following summer Dr. Cheng hired a math specialist from the district to work with me. The math specialist's job was to go over this course of study with me to see if it covers all of the standards in Integrated Math 1, and if it lacked anything we would work on it together until all the standards were covered. After several good discussions and a few modifications, we resubmitted the course of study to my district and it was approved.

Many students that start off going to a City/Community College end up dropping out before they finish. Administrators and teachers for Sacramento City College are trying to improve this statistic. There are several reasons why this percentage is lower than they want. Some of these factors can be controlled and others can not. The faculty at Sacramento City College found out that one of these factors is that the students have so many units to complete that they often give up and get a job that does not require a college education.

To combat this problem Miela Zitelli, a CTE Transitions Coordinator for Sacramento City College, set up a meeting and attended along with Michael Dixon, Professor in the Computer Science department for this college, and me. Their goal for this meeting was for the students who do well in my high school computer science class to earn both high school and college credit. After meeting with me for a while they came up with the requirements for this to happen: I would need to rewrite the course of study for my district to align with the one for Sacramento City College. Once this is done and approved, Michael Dixon agreed that if I am the teacher of this class, all of the students that earn an A grade will get both high school and college credit. A week later I found out that they made this same agreement with another teacher in my district. I reached out to this teacher, we rewrote this course of study together, and it just got district approval in June of 2016.

Creating and Managing Math Practice Programs — Total Hours: 500

Dates: Summer of 2012 to Present

I have written and continue to make programs that give students practice on problems from topics covered in Pre-Algebra, Math 8, Algebra 1, Integrated Math 1, Integrated Math 2 and Algebra 2 for teachers that are using the UC Davis C-STEM curriculum. All of these programs generate different problems made up of controlled random numbers. When the program is first started, it prompts the users on how many problems to do. For each problem the program generates, the students solve/simplify it and input their answer into the computer. The students can input the answer in integer form or fraction form. After the answer is inputted the program tells the student if the problem is correct or not and keeps track of how many correct answers the student earned. After each student finishes the set number of problems they were assigned, it outputs the percent they got correct with an encouraging message.

The first practice program I wrote was for simplifying fractions. When this program starts up it prompts the user to choose what to practice: reducing fractions, adding fractions, subtracting fractions, multiplying fractions, dividing fractions, adding and subtracting fractions, or all operations on fractions. Once the students input this choice, they input how many problems to do. Then the program gives each student their own unique problem to simply and a place to input their answer. After each answer is simplified, the program tells them if it is correct or not and gives the student their next problem until the student finishes the last problem. Then, the program outputs how many they got correct, their percent correct and an encouraging message.

The remaining practice programs for practicing standards in Pre-Algebra is made up of two different programs for operations on integers as well as one for order of operations. The first integer program focuses on adding and subtracting integers and the other integer program also includes multiplication. The program for order of operations includes expressions involving exponents and parenthesis. Just like the program for practicing operations on fractions, these programs output controlled random problems for each student to solve, input their answer and provide feedback.

The next set of practice programs are for solving equations. This is made up of two different programs where the first one only has equations that take three steps or less to solve with at most one fraction in it. The next practice program creates programs that require more than three steps to solve and can be made up of more than one fraction.

For the students to work on problems involving lines, I created five different computer programs. The first of these pratice programs is simply for students to practice finding the slope of the line through two points. A more advanced practice program includes problems for finding the slope of the line that is: parallel to a different line, perpendicular to a different line, when given its x-intercept and a different point, and when given its y-intercept and a different point. The third program for practicing problems involving linear equations gives the students a problem in standard form, point slope form, or one with the variable *x* on the "wrong" side and has the students rewrite the equation into slope-intercept form. After the students have done this, they input the answer they have for slope and the y-intercept into the program. Then the student gets feedback on their answer and receives the next problem. The last two programs in this set is for students to practice finding the equation of a line when given some information about it. The first of these two programs tells the students one of two facts: the slope of the line and a point through it or two points on the line. The other program for finding the equation of the line also includes problems that give: a line in standard form that is parallel to it and a point on it, a line in standard form that is perpendicular to it and a point on it, when given its x-intercept and a different point on the line, and when given its y-intercept and a different point on the line.

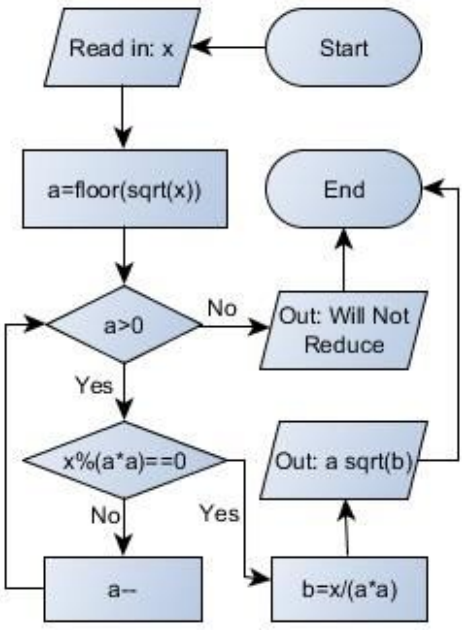
```
>ch -u "practiceSystems3.ch"
Enter the number of problems you want to do:
2

#1: Solve:
          a-2b=-4
          a- b=-1
Enter your answers:
  a = 2
  b = 3

*** Great Job ***
```

The standard that is often covered directly after lines is systems of linear equations. I have created five different computer programs to give students in Algebra 1 as well is in Integrated Math 1 practice on this standard. The first of these five program focuses only on linear systems where both equations are in standard form, at least one of the coefficients is 1, and all of the answers are integers. The second of these programs is very similar to the first, but all of the coefficients are not 1 or -1. The third of these programs

produces problems in different formats: e.g., one equation in standard form, the other in slope intercept form, both equations in slope intercept form, and one equation is solve for x and the other in standard form. The second to last of these programs provides more difficult problems because the answers to each system are fractions. The last of these programs is similar to the one before it with the only exception that the students enter their answer as a decimal.

Operations on quadratic polynomials is a large topic in Algebra 1 and Integrated Math 2. I have created practice programs for multiplying monomials, factoring quadratic expressions and solving quadratic equations by factoring. Just like all of the pratice programs I have created, each student is given their own unique problems that they have to simplify/solve and input their answer as an integer or a fraction, and the computer checks to see if the answer is correct. Once each student has finished all of their problems, all of these programs output how many they got correct, their percent and a positive message.

The last set of these pratice programs, which is often the last topic covered in Algebra 1, is solving quadratic equations that do not factor. The students must first know how to simplify expressions involving square roots before they can solve any quadratic equations. Because of this, I created a program where the students practice simplifying radical expressions. Once the students have mastered this skill, they use one of the two programs I created for practicing solving equations using the quadratic formula. The first of these programs only gives students quadratic equations in standard form, while in the last of these programs the problems are given in several different forms. In order to simplify the answer when using the quadratic formula, I had to design and implement an algorithms that reduces square roots. The flowchart for this code can be found on the right.



All of code to these programs can be found at http://mrdagler.com/practice_programs.html. In order to run them you will need to download and install the ChIDE which can be found at https://www.softintegration.com/docs/ch/chide/.

Robot Demonstration at UC Davis Picnic Day — Total Hours: 10

Date: 04/16/2016

UC Davis Picnic Day is an annual open house where over 200 different events occur throughout the campus including exhibits, shows, competitions, demonstrations, entertainment, animal and athletic events. I was part of an activity for the College of Engineering where some of my students and I used programming to have several robots compete in a soccer match. Once a new observer came by, I or one of my students showed him/her some basic programming and they were ready to join the match. At several times throughout the day we had over 40 people with an age range of about five to over 65 observing and programming the robots. Each observer left our activity with an increased interest and a better understanding of robots and coding.

Created and Managed a Program that Produced Data for QEIA Funds. **Total Hours: 150**

Dates: 11/05/2009 to 06/15/2015

The school I work at was awarded QEIA funds (Quality Education Investment Act) from the 2008/2009 to the 2014/2015 school year. One of the requirements for this grant is that the number of students in the core class (mathematics, science, social science, and English) have to stay under 27. Each month of the school year we had to prove this was true. This does not seem very hard to do, but the school I work at has around 500 classes each semester and simply running a monthly list of class sizes, identifying which ones are core classes and showing each of them are under 27 is a large task. Another requirement for this grant is to show that class sizes in these core classes average under 22.7 by grade level. This involves a more than adding and dividing. In order to do this calculation you need to define what class is a 9^{th} grade class, 10^{th} grade class, 11^{th} grade class, and 12^{th} grade class. In some cases this distinction is obvious. An example would be Senior English, where every student in there would be in 12th grade. However, what about a biology class which can be made up of 7 freshmen, 11 sophomores, 8 juniors and 5 seniors? To handle classes that are made up of more than one grade level, I had to use the mode so this biology class would be consider a 10^{th} grade class.

Since this is a large task that needs to be done several times, the administration team for my school asked me to create and manage a computer program which would handle all of these calculations and generate useful reports. First, I had to extract the data from the SIS (student information system) that we were using at the time, and from there I wrote a program in C++ that reads in this data, performs the necessary calculations, and writes the output to a file that can be formatted using a spreadsheet program. After I created and tested this program, I had to upkeep it when a new course was added to the school, and one big update was required when we switched our SIS from Zangel to Infinite Campus.