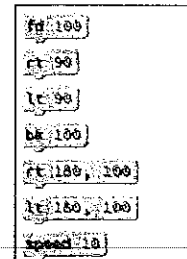


The Power of Teaching Mathematics with Code

by Jeramie Oliveira, Simi Valley USD
jeramie.oliveira@simivalleyusd.org



In May 2016, our Director of Curriculum and Instruction, Pam Castleman, discovered two hybrid computer science/mathematics classes created by the University of California Curriculum Integration (UCCI) project. My colleague, Dan Shuster at Royal High School, and I were approached as potential teachers for the Geometry with Computer Visualization and Simulation and Algebra 2 for the 21st Century mathematics/programming hybrid courses. Dan thought we should call it “Math with Code” for short.

I started teaching the geometry class in August 2016 and immediately saw the benefits of implementing code in mathematics. Students liked the different approach to learning mathematics and enjoyed the hands-on projects. Most of all, they appreciated that I admitted that I was not a programming expert and needed their help to teach the class.

That was the game changer. Students started doing their own research and shared ways to approach the project ideas that I presented. Many students showed me programming tips, tricks, and ideas and improved my suggested code, while still applying the mathematical skills of my class. Even the students who did not sign up for a code-based class and had never coded before enjoyed the class. Most of the students even signed up for the Algebra 2

version that I am teaching this year and one of my students returned to the Geometry class as a peer tutor.

Teaching mathematics with code provides a unique learning experience for students. In this kind of class, students have multiple opportunities to “teach” mathematics. Think about this for a minute. When a student writes code, the student is “teaching” the computer how to complete a task and uses all of the Mathematical Practice Standards naturally while writing the programs. Here is a brief description of how each Mathematical Practice Standard is used when teaching mathematics with code.

◆ Make sense of problems and persevere in solving them

When students write in code, they immediately see if their program is successful at completing the task. The code is rarely correct the first time, just like some complex mathematical calculations. As a team, students look for the errors and attempt to fix their code. They often repeat the process (sometimes more than 20 attempts) to get their code to function properly, and they do not give up. They will ask for guidance, research online, and ask other teams to get their code to work properly. All this happens while they are trying to “teach” a

computer how to do the mathematics.

◆ **Attend to precision**

Mistakes made with code are apparent as soon as the code is run; however, finding the mistake can be an arduous task. Teaching students how to find an error in code follows much of the same logic as finding a mathematical error in pages of work. While many students lack the desire to find mathematical errors, finding an error in their code becomes an engaging quest.

◆ **Model with mathematics**

Every program that a student writes in code is a computer model. The essence of computer programming is to create a model that can return various outputs based on user input. Using mathematics to teach the logic of programming brings the modeling to the mathematics and vice versa.

◆ **Use appropriate tools strategically**

Computer programming has many commands to choose from just as mathematics has many problem-solving techniques. There are *if* statements, *while* and *for* loops, user-created functions, etc. that students use to turn mathematics properties, theorems, and formulas into working programs. Since every command is a tool that makes a program function, students must use those tools effectively for efficient program code.

◆ **Look for and make use of structure**

Writing code is all about structure and flow. Students soon learn that user-defined functions have to be placed in certain locations. Code runs in a linear manner and loops follow instructions. Just like individual steps in solving a mathematics problem, pieces of code can run independently. However, it is how we put those steps together that generates completed solutions and programs.

◆ **Look for and express regularity in repeated reasoning**

Programmers do not rewrite the same code over and over again. Instead, they save functions that can be reused in other programs that require the same calculations. By learning this skill and creating a library of pre-coded functions, students are able to perform more

complex tasks. This process is the same as learning mathematical theorems and formulas that help students perform more complex tasks.

◆ **Reason abstractly and quantitatively**

Coding is a very abstract process. Programmers should do their best to use comments and descriptive variables so others who read their code can understand their reasoning. However, the users of most computer programs want a quantitative output. A calculator is not useful because of the programmed code; it is useful because the code returns a number. Writing code is both abstract and quantitative.

◆ **Construct viable arguments and critique the reasoning of others**

Coding projects are better done as a collaborative effort and should not be done as individual projects. When students work together to write, test, and create mathematics-based code projects, they are talking, experimenting, and engaged in the mathematics and the coding process.

Project Examples

My goal is to use mathematics as a way to introduce computer programming to my students and to use computer programming to reinforce mathematics skills learned in class. By putting the power in the hands of students to create their own tools to model mathematics, students will often find ways to do things that are not anticipated. I offer extra credit to students who teach me something new in their code. Teachers will learn a lot from students in the first few years of teaching Mathematics with Code since many students will add their own flair and their own signature to their programs. As an added bonus, students learn 21st century skills of teamwork, collaboration, and problem solving.

Below are two examples of coding projects that my students will be completing in the fall of 2018.

◆ **Transformations using code**

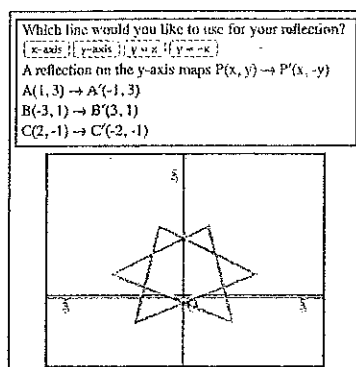
The Transformation Project is one of the first projects in Geometry with Code. Students will complete this programming project as a team,

Continued on page 20 > >

using the directions provided in the program "shell" at bit.ly/MWCTransformationsShell. The code students develop will allow for a triangle to be translated, rotated, or reflected.

The complete teacher-created code is also located at this site for your reference. To see the working program and try it out, go to bit.ly/MWC-Transformations.

The working program can be used to demonstrate translations, rotations, or a reflection on a triangle for students.



◆ Quadratics using code

In the Graphing Quadratics Project, students use some of the code that they created in the Geometry course. Students that do not take the Geometry with Code class are given these code functions to keep up with the new code requirements. This code asks students to input the a , b , and c values of a quadratic equation,

and will then give the graph, discriminant, and solutions to the given quadratic when students run the program.

The link for the project shell and the teacher-created code is located at bit.ly/MWCCGraphingQuadraticsShell. For teachers who want to run the program, the link is bit.ly/MWCCGraphingQuadratics.

Conclusion

According to the Code.org website, there are 557,903 open computing jobs nationwide, but only 49,291 college graduates have the skills to fill those positions. Teachers do not have to be coding experts to bring code to their classrooms. They simply have to imagine where coding will take their students.

References

Alg2/Shells/GraphingQuadratics. <http://bit.ly/MWCCGraphingQuadraticsShell>; <http://bit.ly/MWCCGraphingQuadratics>.

Code.org. <https://Code.org>.

Geo/Shells/SplitS-Transformations. <http://bit.ly/MWCTransformationsShell>; <http://bit.ly/MWCTransformations>. 