STEMJOBS℠
DO WHAT YOU LOVE.

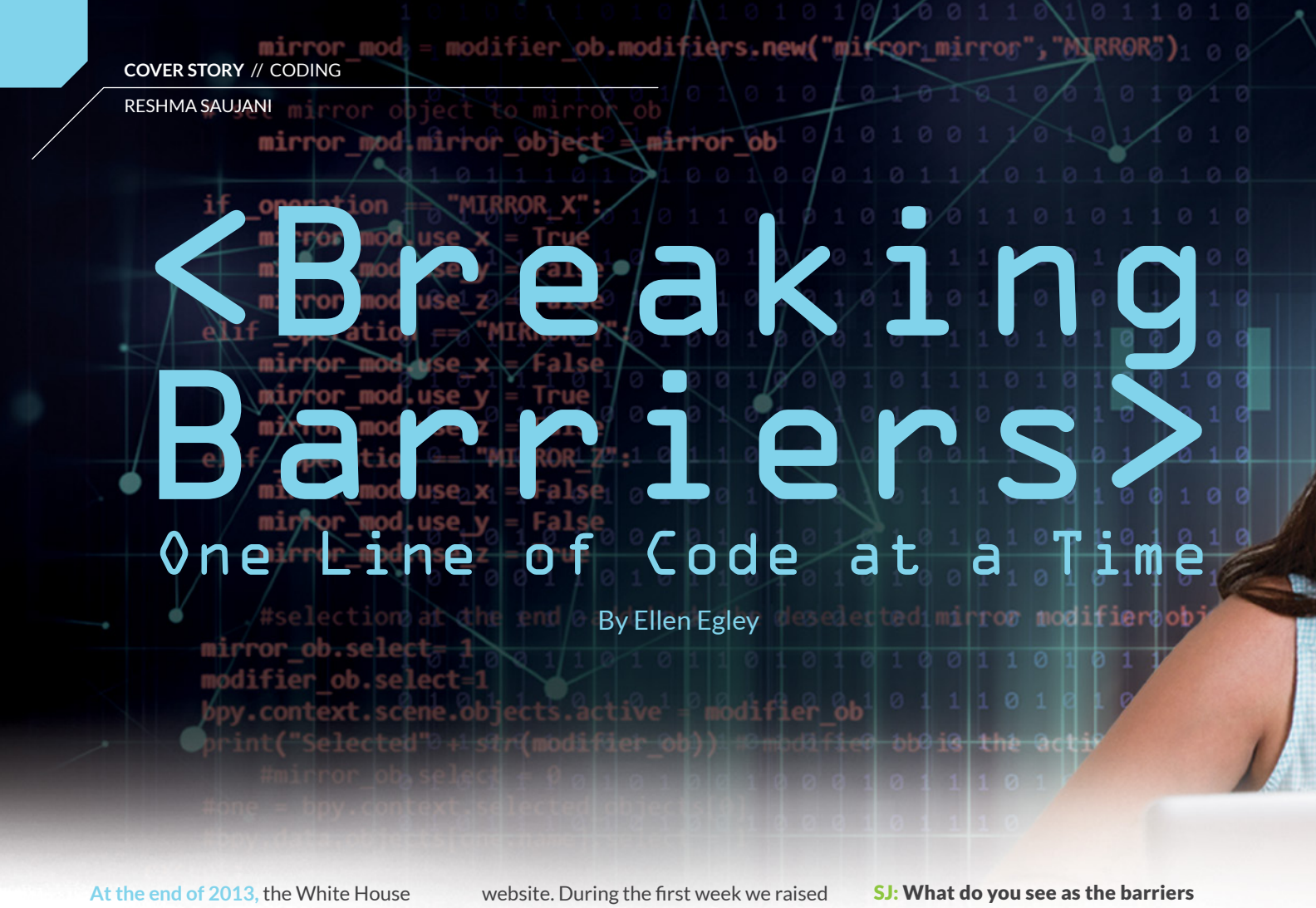
5-E CLASSROOM STEM ACTIVITY:
CODING FOR A CAUSE

Alexandra Owens

<Breaking Barriers>

One Line of Code at a Time

By Ellen Egley



At the end of 2013, the White House predicted there would be 1.4 million computer-science-related jobs available by the year 2020, but that the U.S. would only produce about 400,000 computer science graduates to fill those jobs. That means we'll only have enough qualified people to fill less than 30% of those jobs! Worse yet, only one out of every four of those jobs will be filled by women.

This disturbing trend led Reshma Saujani to start a nonprofit organization geared toward teaching girls in middle and high school how to code through afterschool programs and summer camps. *STEM Jobs* talked with Reshma about Girls Who Code, what inspires her, and what the future of tech holds for those brave enough to take the leap.

STEM JOBS: What inspired you to start Girls Who Code?
RESHMA SAUJANI: I'm an unlikely person to be leading this charge—I am not a coder. In 2010, I decided to run for United States Congress in a democratic primary against an 18-year incumbent because I thought that was a great idea. I had a 1% chance of winning, a 1,000-page policy book, and the only thing my friends and I knew how to do was build a website. During the first week we raised $50,000 from Indian aunties that were just so happy an Indian girl was running. On Election Day, I swore I was going to win and as we watched the election returns come in I got only 19% of the vote. I was broke, humiliated, and I had no contingency plan. But as I went to bed that night the faces I kept seeing over and over again were actually the ones I had never seen on the campaign trails.

During the campaign, I would visit schools and see armies of boys learning to code, training to be next Mark Zuckerberg or Steve Jobs. And I thought to myself, "Where are all the girls?" This question became my obsession. It didn't seem right to me. At a time when women are a majority of college graduates and close to a majority in the labor force, where are we in tech, the industry shaping our collective future?

SJ: What is the mission of Girls Who Code?
RS: We're a national nonprofit organization dedicated to closing the gender gap in technology. It's a solvable problem, one that I think we can achieve within our lifetime.

SJ: What do you see as the barriers between girls and careers in computer science?
RS: I think culture is a huge barrier. But it hasn't always been this way. In the 1980s, 37% of computer science graduates were women. Steve Jobs' original MacIntosh team had more women than most tech companies today. So what changed? In the 1980s, personal computers came out and were marketed as a game for boys. That narrative got picked up in movies like *Revenge of the Nerds* and continues today in TV shows like *Silicon Valley*. The image of a programmer these days is of a boy in a hoodie in a basement alone and girls look at that and say "no, thanks." So we need to change pop culture and the image of what a programmer looks like and does. Which is why we're releasing a first-of-its-kind 13-book series with Penguin as an invitation for girls everywhere to learn to code and change the world. One of the best ways to spark girls' interest is to share stories of girls who look like them. You can't be what you can't see, so we created five relatable characters to feature in every book. Lucy, Maya, Sophie, Erin, and Leila represent the diversity and range

RESHMA SAUJANI
CEO & FOUNDER OF GIRLS WHO CODE
DEGREES: BACHELOR'S IN POLITICAL SCIENCE, MASTER'S IN PUBLIC SERVICE, JURIS DOCTOR (LAW DEGREE)
YEARS IN THE INDUSTRY: 5
STEM TYPE: INTEGRATOR

"TODAY ALONE THERE ARE 500,000 OPEN JOBS IN COMPUTING. THESE ARE THE BEST-PAYING JOBS IN THE COUNTRY."

of backgrounds of our girls who code across the United States. We hope that girls will be able to seem themselves in these characters and as a result see themselves as coders.

SJ: You released two new books, *Girls Who Code* and *The Friendship Code*, at the end of August. What is the inspiration behind and the goal of each?
RS: After hearing from parents across the country that they wanted their daughters to learn to code, I went on Amazon to see what books were available and didn't see anything for girls and coding. Thus the idea of *GWC* books were born.

The nonfiction book contains explanations of computer science and coding principles using real life examples to get readers to start to think like a computer scientist, including examples of projects from our very own girls who code. *Girls Who Code: Learn to Code and Change the World* is the Girls Who Code approach to computer science in graphic novel form—we have art on every page to make the content relatable and visually engaging.

The fiction book series really stems from the idea that you can't be what you can't see, so we wanted to create a series of books with relatable characters and role models from all walks of life. Each of the books is told from the perspective of one of the five characters we've created so that all girls can see themselves as coders, whether they like sports or baking or fashion!

SJ: Did anything surprise you as you began to work with girls through Girls Who Code?
RS: What has amazed rather than surprised me is how imaginative and innovative girls are when they learn to code; they become change makers.

SJ: What advice do you have for teens interested in computer science and coding? Why do you think this is a career path worth pursuing?
RS: This is absolutely a career path girls should pursue! Today alone there are 500,000 open jobs in computing. These are the best-paying jobs in the country. These jobs pay twice that of the average private sector job and are at the center of innovation and growth. And they're not just in Silicon Valley. In fact, 91% of open software and computing jobs are outside of Silicon Valley. In Michigan there are 15,000 currently open jobs. In Virginia, 34,000. In Ohio, 15,000. Georgia, 20,000. These jobs aren't just in tech either. Sixty-seven percent of computing jobs are in retail, banking, transportation, entertainment, agriculture, manufacturing, even government! I'd also ask them to think about a problem they want to solve. If you learn to code, you can solve that problem.

As for advice, just be brave and go for it. In our society, we train boys to be brave—to throw caution to the wind and follow their passions. And we train girls to be perfect—to please and play it safe, to follow the rules, and to always get straight A's. The result? Girls are kicking you-know-what in the classroom, but falling behind in the real world. Because in the real world, success is a product of bravery, not perfection. If we don't start teaching girls to be brave, they are going to miss their chance to code the future in Silicon Valley, to build the future in the C-suite, and to legislate the future on Capitol Hill. So jump in and start learning, and don't worry about failing.

5-E CLASSROOM STEM ACTIVITY: CODING FOR A CAUSE

Here are some ideas for how high school teachers could use this story as a launching point for integrated STEM learning. Our activities follow the 5-E Learning Cycle Model.

Part 1: Engage

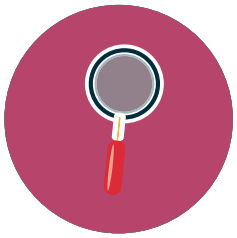
1. Hold a class discussion with students about apps on their phones.
   - **a.** What is your favorite app?
   - **b.** Why is it your favorite?
   - **c.** What apps do you use to help make your life easier?
   - **d.** What apps do you use to help you in school?
   - **e.** How are apps designed?
2. Have students read the article "Breaking Barriers—One Line of Code at a Time" in *STEM Jobs* magazine. Discuss the following questions:
   - **a.** Why does a gender gap exist in the computer science field?
   - **b.** What are the goals of Girls Who Code and how will they address the gender gap?
   - **c.** What problems would you like to solve by coding an app?
3. Show students the website "MIT App Inventor" found at **edu.STEMjobs.com/teacher-resources**. Using this free website, students can learn basic coding to create an app for an Android phone. Students can use any web browser to code, and if an android phone is not available, you can download a phone emulator to test created apps.

Part 2: Explore

1. Break students into pairs. They may each work separately on their own device, or work together on a shared device.
2. Direct students to learn how to code using App Inventor by completing a series of beginner tutorials found at **edu.STEMjobs.com/teacher-resources**. Recommended tutorials include Talk To Me and PaintPot. Allow students to explore additional tutorials that they would like to complete in order to learn about other features, such as the location sensor, accelerometer, or file sharing that would be applicable to their project.
3. Using what they have learned in the tutorials, ask students to create a basic app to practice these skills. They should test their app using a phone or emulator to ensure the coding is correct.

Part 3: Explain

1. Present students with a challenge to design an app that will solve a problem that students face at your school. In order to determine both the needs of the school and potential impact, have students collect data through means such as conducting a survey, researching statistics, or interviewing the administration.
2. Once each pair has identified the problem they would like to solve, they will work together to design an app that can help address that problem. Provide plenty of time for students to create their app design using App Inventor. Learning is an iterative process, so periodically have pairs show their progress to another group to get constructive feedback and make improvements.
3. Once apps are complete and have been thoroughly tested, pairs will create a short presentation of their design to share with the class. Have the groups present results of their data collection, the problem they decided to solve, and their app design. Each member should share the information and coding that they were responsible for. Any data collection devices used should be presented and explained. Students should share their device, as well as explanations using presentation software such as PowerPoint or Google Slides. Students must include a mathematical representation of their data collection to support how their app will address the school's needs and potential impact, such as data tables or graphs.
4. After each presentation, ask audience members to provide constructive feedback regarding their app design.



Part 4: Elaborate

1. Once presentations are complete, have groups reconvene and present a twist. Ask students to consider how a second version of this app could be designed for a local elementary school. Students should consider how features may need to change, or what should be added.
2. If time allows, have students create a second version of their app. If there are time constraints, ask students to brainstorm and create a pitch describing their modified app.
3. Have groups share their new version of the app or marketing pitch with the class for constructive peer critique.

Part 5: Evaluate

Students will be evaluated for their app design and presentation using the following rubric.
Provide the rubric at the beginning of the lesson to clarify expectations and objectives.
Each group will be graded, therefore all students in the group will receive the same score.

Scoring Rubric

____ /20 App Design
Did the app have working components? Was the app coded correctly? Is there sufficient data to support the needs of the school and the potential impacts that the app may have? Was a second version of the app modified for the needs of an elementary school?

____ /20 Participation
Did each student contribute to the overall project? Did each student assist in coding and testing the app?

____ /10 Presentation
Did the app meet a need of the school? Did the presentation include a working app and mathematical representations of the measurements? Was the presentation clean and easy to understand?

____ /50 Total

Standards Addressed:

Common Core State Standards – Math

CCSS.MATH.PRACTICE.MP1 Make sense of problems and persevere in solving them.
CCSS.MATH.PRACTICE.MP2 Reason abstractly and quantitatively.
CCSS.MATH.PRACTICE.MP4 Model with mathematics.
CCSS.MATH.CONTENT.HSS.ID.A.1 Represent data with plots on the real number line (dot plots, histograms, and box plots).

Common Core State Standards – ELA

CCSS.ELA-LITERACY.SL.9-10.1 Initiate and participate effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grades 9-10/11-12 topics, texts, and issues, building on others' ideas and expressing their own clearly and persuasively.
CCSS.ELA-LITERACY.RST.9-10.1 Cite specific textual evidence to support analysis of science and technical texts, attending to the precise details of explanations or descriptions.
CCSS.ELA-LITERACY.RST.9-10.4 Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 9-10 texts and topics.
CCSS.ELA-LITERACY.RST.9-10.7 Translate quantitative or technical information expressed in words in a text into visual form (e.g., a table or chart) and translate information expressed visually or mathematically (e.g., in an equation) into words.
CCSS.ELA-LITERACY.RST.11-12.7 Integrate and evaluate multiple sources of information presented in diverse formats and media (e.g. quantitative data, video, multimedia) in order to address a question or solve a problem.
CCSS.ELA-LITERACY.RST.11-12.9 Synthesize information from a range of sources (e.g., texts, experiments, simulations) into a coherent understanding of a process, phenomenon, or concept, resolving conflicting information when possible.
CCSS.ELA-LITERACY.SL.11-12.1 Initiate and participate effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grades 9-10/11-12 topics, texts, and issues, building on others' ideas and expressing their own clearly and persuasively.
CCSS.ELA-LITERACY.SL.11-12.5 Make strategic use of digital media (e.g., textual, graphical, audio, visual, and interactive elements) in presentations to enhance understanding of findings, reasoning, and evidence and to add interest.

Next Generation Science Standards

HS-ETS1-2. Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
Science and Engineering Practices
Constructing Explanations and Designing Solutions. Design a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.
Disciplinary Core Ideas
ETS1.C: Optimizing the Design Solution. Criteria may need to be broken down into simpler ones that can be approached systematically, and decisions about the priority of certain criteria over others (trade-offs) may be needed.
HS-ETS1-3. Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics as well as possible social, cultural, and environmental impacts.
Science and Engineering Practices
Evaluate a solution to a complex real-world problem, based on scientific knowledge, student-generated sources of evidence, prioritized criteria, and tradeoff considerations.
Disciplinary Core Ideas
ETS1.B: Developing Possible Solutions. When evaluating solutions, it is important to take into account a range of constraints, including cost, safety, reliability, and aesthetics, and to consider social, cultural, and environmental impacts.
Crosscutting Concepts
Influence of Science, Engineering, and Technology on Society and the Natural World. New technologies can have deep impacts on society and the environment, including some that were not anticipated. Analysis of costs and benefits is a critical aspect of decisions about technology.

ISTE Standards

1d Students understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.
3d Students build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions.
4a Students know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.
4b Students select and use digital tools to plan and manage a design process that considers design constraints and calculated risks.
4c Students develop, test and refine prototypes as part of a cyclical design process.
4d Students exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.
5b Students collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making.

Texas Essential Knowledge and Skills- Math

A.1.A apply mathematics to problems arising in everyday life, society, and the workplace.
A.1.B use a problem-solving model that incorporates analyzing given information, formulating a plan or strategy, determining a solution, justifying the solution, and evaluating the problem-solving process and the reasonableness of the solution.
A.1.D communicate mathematical ideas, reasoning, and their implications using multiple representations, including symbols, diagrams, graphs, and language as appropriate.
A.1.E create and use representations to organize, record, and communicate mathematical ideas.
A.1.F analyze mathematical relationships to connect and communicate mathematical ideas
A.1.G display, explain, and justify mathematical ideas and arguments using precise mathematical language in written or oral communication.

Standards Addressed (Cont.):

Texas Essential Knowledge and Skills- Technology

FCS.1.D create algorithms for the solution of various problems
FCS.1.G design creative and effective user interfaces
FCS.2.A seek and respond to advice from peers and professionals in evaluating problem solutions
FCS.2.B debug and solve problems using reference materials and effective strategies
FCS.4.E demonstrate coding proficiency in a contemporary programming language by developing solutions that create stories, games, and animations
FCS.4.G demonstrate an understanding of and use variables within a programmed story, game, or animation
FCS.4.I demonstrate an understanding of and use sequence within a programmed story, game, or animation
FCS.4.J demonstrate an understanding of and use conditional statements within a programmed story, game, or animation
FCS.4.K demonstrate an understanding of and use iteration within a programmed story, game, or animation
FCS.4.L create an interactive story, game, or animation
FCS.5.E discuss the impact of computing and computing related advancements on society
FCS.6.E describe, compare, and contrast the differences between an application and an operating system
CSI.4.A use program design problem-solving strategies to create program solutions
CSI.4.B define and specify the purpose and goals of solving a problem
CSI.1.4.C identify the subtasks needed to solve a problem
CSI.1.4.D identify the data types and objects needed to solve a problem
CSI.1.4.F design a solution to a problem
CSI.1.4.H identify and debug errors
CSI.1.4.J debug and solve problems using error messages, reference materials, language documentation, and effective strategies
CSI.1.4.M create program solutions that exhibit robust behavior by understanding,
avoiding, and preventing runtime errors, including division by zero and type mismatch
CSII.1.A use program design problem-solving strategies to create program solutions
CSII.1.C follow the systematic problem-solving process of identifying the specifications
of purpose and goals, the data types and objects needed, and the subtasks to be performed
CSII.2.A use the principles of software engineering to work in software design teams, break a problem statement into specific
solution requirements, create a program development plan, code part of a solution from a program development plan while
a partner codes the remaining part, team test the solution for correctness, and develop presentations to report the solution findings