Lessons Learned

Teaching programming to young beginners

Barbara Miner, Ph.D.
Retired engineer, Retiring educator

Personal Experience

Teaching ~1900 **girls** (all-girl classes), ~900 students in co-ed classes

Age range 2nd to 8th grade. Mostly 4th – 6th

Lesson Learned

After the first year of the program, I stopped offering programming classes to 2nd graders. All truly unsuccessful girls were 2nd graders.

Experience

I was an engineer for 25 years at Intel and I have the lived experience that more “girl engineers” are needed. There is data that more diverse teams are more creative and can be more productive. When I took early retirement from Intel, I became a social venture partner at Saturday Academy. My project was to create a girls’ program.

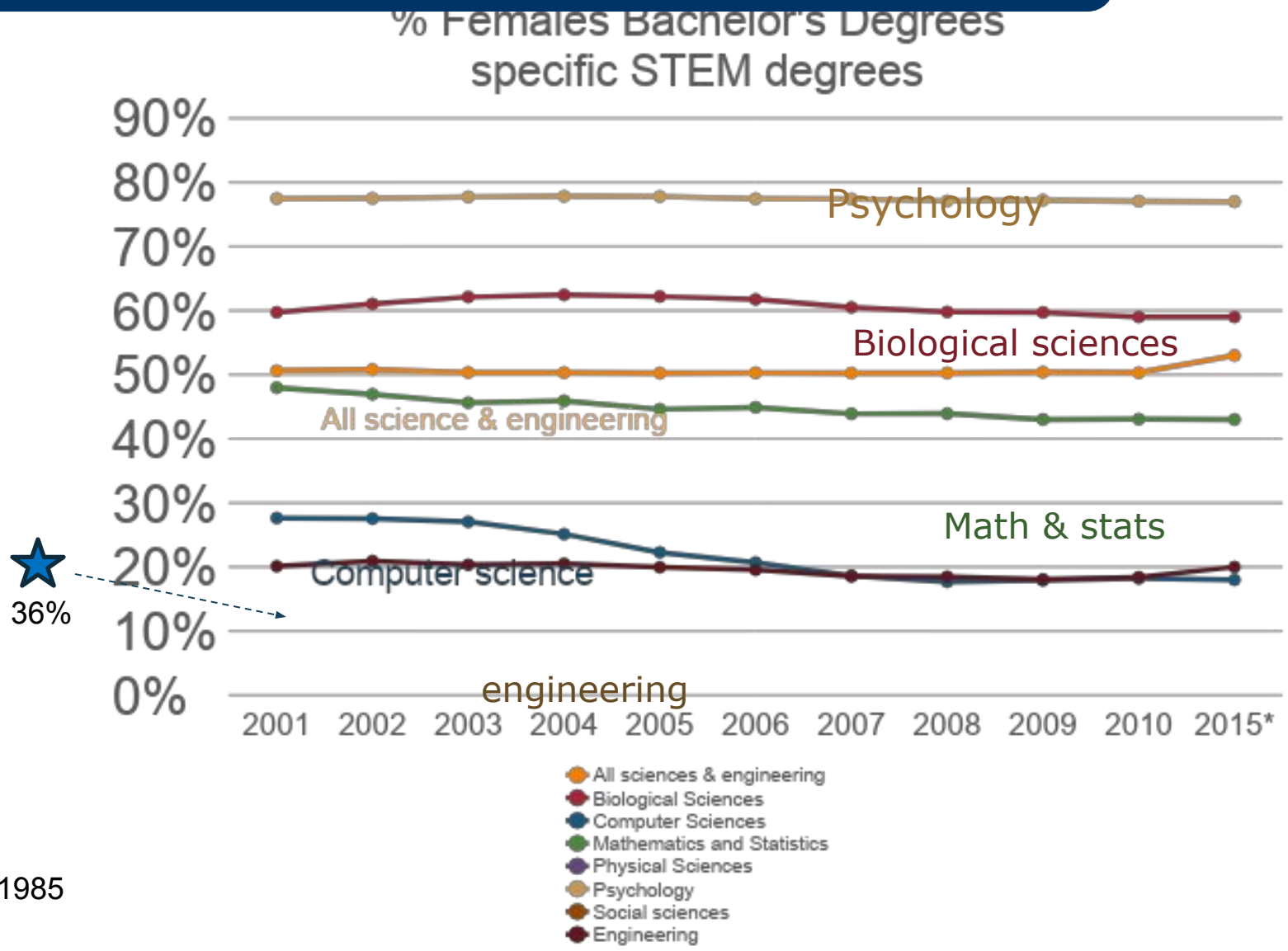
I thought it would be in science. Then I looked at the data.

Many STEM fields have achieved gender parity!

But female participation in engineering and computing is about 19%.

My first male boss introduced me as “This is my Girl Engineer”

Problem Statement - society



36%

1985

data from 2013_nsf_women_minorities_science_eng_full.pdf with 2015 update and 1985 data, also from nsf reports

Focus on Girls*

My research was on what methods are best for teaching and inspiring Girls in engineering (mostly programming).

However, all these methods are useful for reaching any student who is uncertain of success -- including children who do not get good grades, who have no familial connections to engineers, who have no computers at home, or children who feel that they do “not belong” in this group.

Research on Best Practices for Teaching Girls

- Increased post-assessment of ***knowledge in programming correlated with increase in confidence in succeeding*** in computer science.
- Students with programming experience are more confident and more successful *in introductory courses* than their inexperienced peers and more likely to persist.
- Girls enter college with fewer programming experiences.

Skills needed

Girls Engage Technology

- Provide ***positive engineering*** experiences.
- Confidence through Competence
- Lower the barriers to participation:
  - Cost: Classes are supported by grants and free to all girls.
  - Location: Classes are mobile and can come to your girl group
  - Intimidation: taught in all-girl groups
- Target age group is pre-middle school with a small number of teens as near-peer mentors.
  - Students choose their identity and narrow their interests in middle school. HS students are determining college major and career.

Hands-on classes

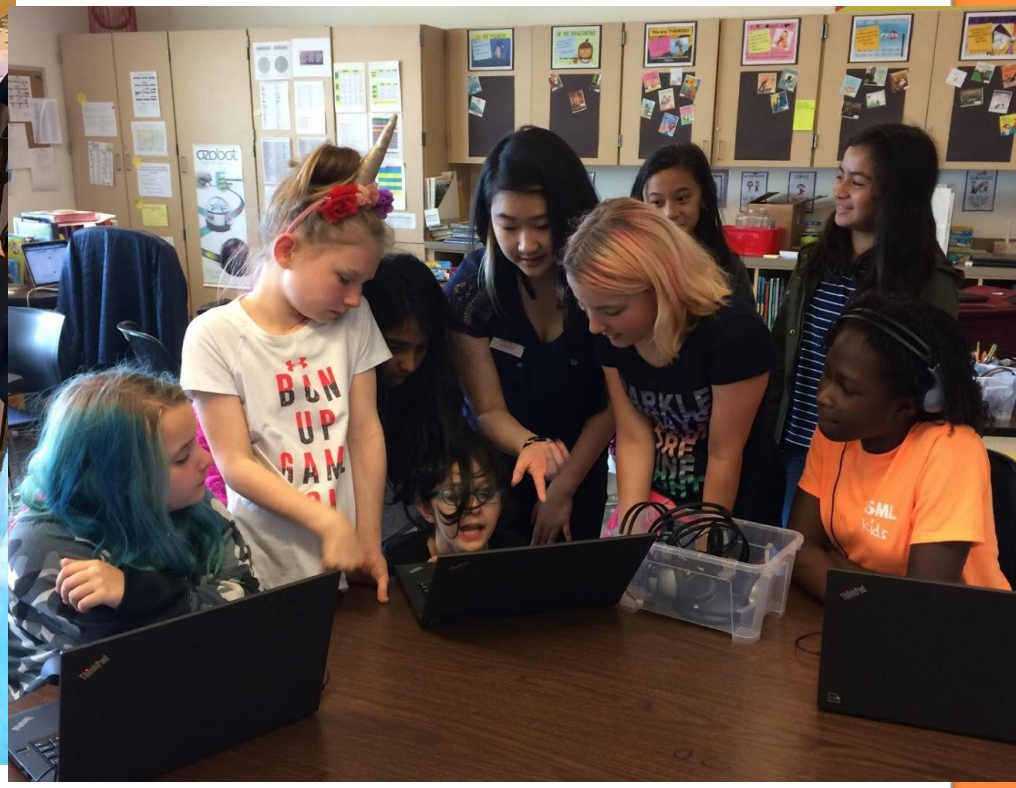
Reality check

Girls Engage Technology **Post-assessment**	**YES**	**maybe**	**NO**
I have **written a computer program**			**X**
I am good at solving problems	X		
I am the kind of person who works well with computers	X		
I would like to take more computer classes	X		
Would you recommend this class to a friend?	X		

- We started ***verbalizing the accomplishment***. "what an interesting program you wrote". "That program has never been written by anyone before."

Many girls like to build things, but *don't see themselves* as engineers.

(bring in the teens)

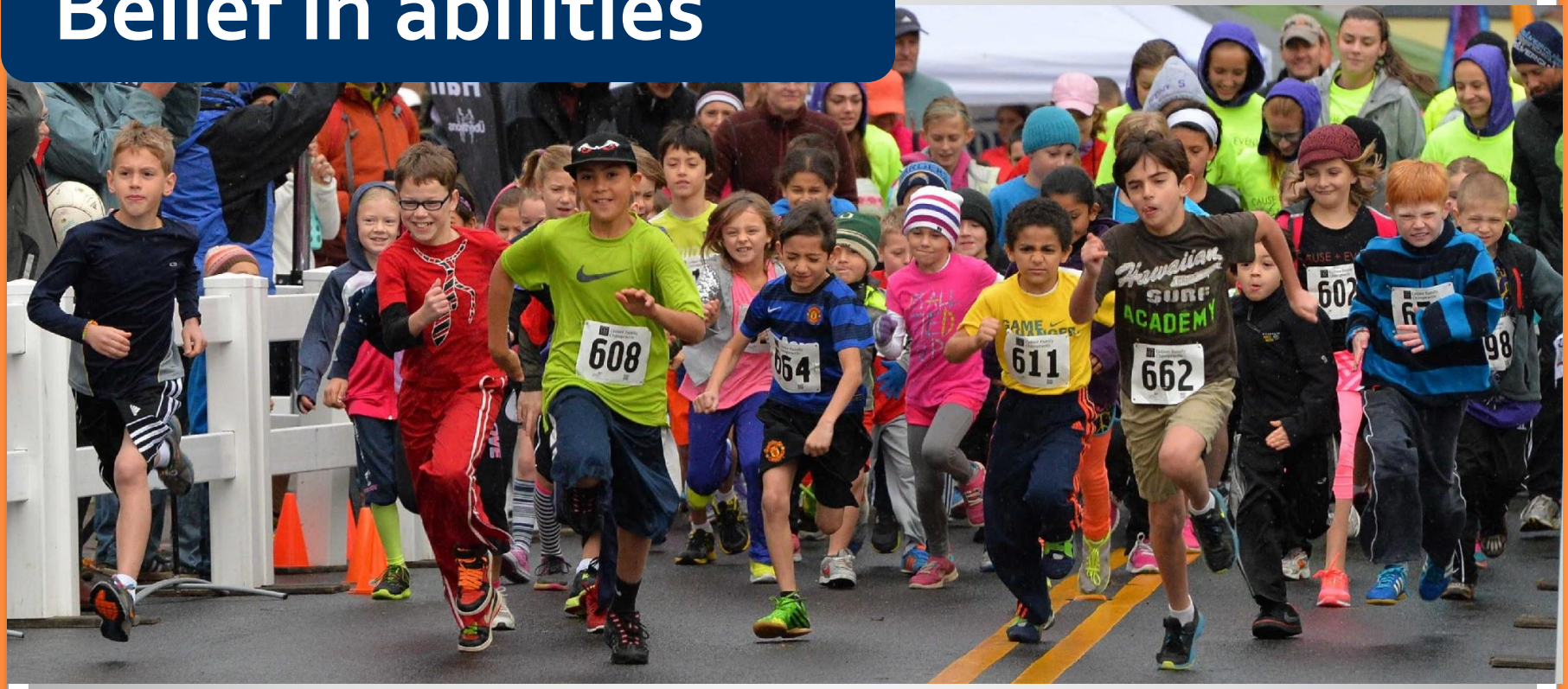


Generation STEM: What girls say about Science, Technology, Engineering and Math 2012

Research on Best Practices for Teaching

- Children's ***beliefs about their abilities*** are central to determining their interest. (both genders)

Belief in abilities



Girls with higher levels of experience in programming reported higher levels of confidence (than those without the experience), but boys reported being confident regardless of experience.

Gender and career-choice process: the role of biased self-assessments. Shelley Correll American Journal of Sociology vol 106 #6 2001. pp 1691-1730

Research on Best Practices for Teaching Girls

- Equal competence does not guarantee equal confidence.
  - 65% of boys claimed their computing skills are more advanced than the girls, while 19% of girls claimed their skills were more advanced than the boys. (all students were in AP computer classes; grades were higher for girls.)
  - After competence, there is a (statistical) gender difference in confidence

Increase confidence

"Still a stranger here: attitudes among secondary school students towards CS."

Methods for increasing confidence

- **Girls in all-girl classes had higher confidence in their skill** than did girls in mixed gender classes. (from a paired comparison of HS students in programming classes. Post-assessment.)
- Deliberately **discourage** students from **intimidating** others in class by showing off their knowledge

Methods for increasing confidence

- Teach that abilities are expandable and improvable
  - "Engineers say "it doesn't work YET". We know it will work
- Do not confuse innate ability with prior experience
- Specific praise for actions.
  - "You have defined the problem perfectly; that makes it easier to solve"
  - "I've watched you try at least 5 conditions before selecting the best one. That's a great method."
- Do NOT assign "smartness" to students.
  - If "you are smart" is associated with a success, then having difficulty is associated with "you are not smart"

Methods for increasing confidence

- Girls have a tendency to ask for adult rescue when a task seems difficult or boring. **Encouragement to persist** goes a long way to address differences in confidence and perceived ability.
  - At the end of the class, the girl must feel that ***the work is her own.***
- Perceived support *from* ***teachers and peers*** correlated to girl' interest in computing courses

Methods for increasing confidence

- Have regular demonstrations; every girl shows her work
  - Role-model praising specific components
  - Some girls need to be encouraged "I think your program is very interesting and that the other girls would like to see it"
- DO NOT ALLOW sharing to the world-wide Scratch community within the class.
  - The goal is for each student to have pride in her work, not to receive "likes".
- Each girl has her own computer and makes her own decisions.
  - Several studies said that working in pa
    tried it and found otherwise.

My experience

Expand their idea of engineers

- INTRODUCTIONS are important.
  - I introduce myself as a retired engineer, mom of 3 kids (and a dog).
  - What is engineering? (building, solving problems, making things work)
  - I say "we need more engineers with girl brains".
  - "We are practicing being software engineers today."
  - Tell true stories of robots at work. What job would you build a robot to do?
    - >90% of girls self-identified as interested in STEM consider that "people" and "making a difference in the world "are important in choosing a career path".

(Generation STEM: What Girls say about Science Technology, Engineering and Math, 2012)

- Teen introductions are golden

saturday academy · get · Girls Engage Technology

Near-peer mentors

Programming Class

- Select an open-ended project with specific programming goals
  - Write the goals for all to see. Leave them up.
  - Write a story with At least 3 characters. Each one has motion and conversation and begins when the green flag is clicked
  - Include one backdrop change
  - At least one sound
  - At least one loop
  - (Conditional Clauses)
  - (x,y coordinates with game)
- Very short hands-on tutorial. "D

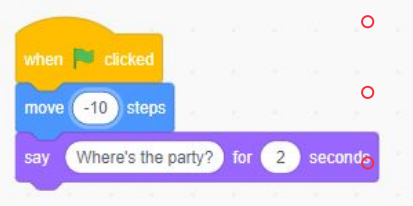
Some classes will go farther than others

Goal is 100% success

- Success = every girl completes the programming goals and has a working product that she is proud of.

Programming Class

- Girls log into scratch.mit.edu
- Type in username and password
- Plug in the mouse *to the USB port*
- Follow along to write a 1st program



- White ovals are VARIABLES (say it with me).
- Test the program (engineers test changes)
- Switch positions of move and say commands. Test.

Quick success wins over some anxious kids

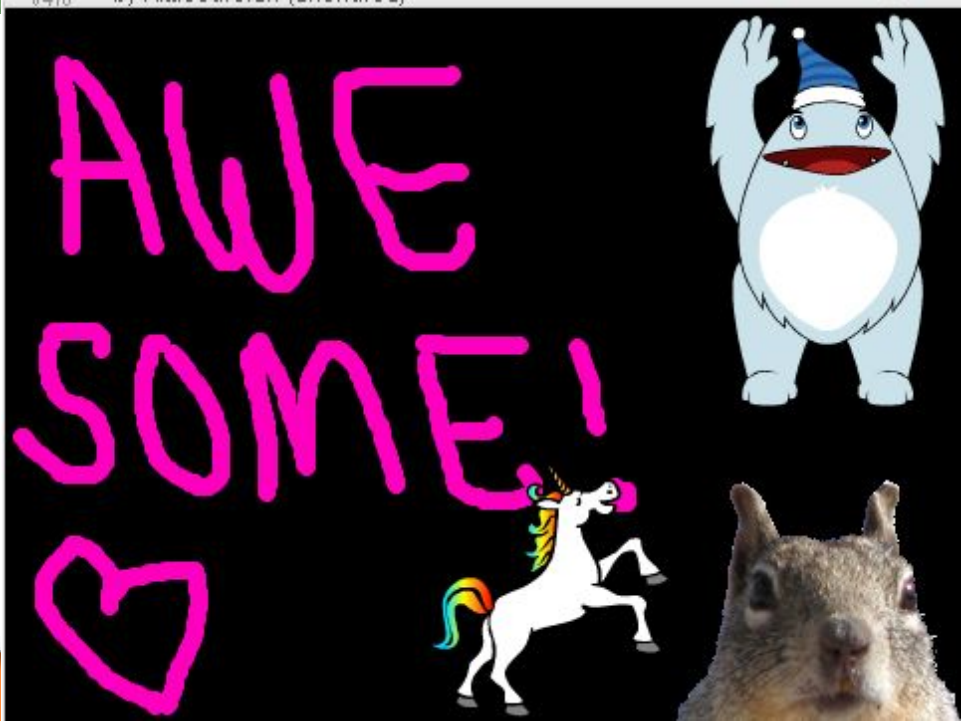
- We look at characters and backgrounds and every girl decides on a story ***she wants to write***.

Interest in topic increases engagement

- Ownership of the selection of the program she will write increases engagement
  - I would be quite bad at knowing what interests 10-year old girls
- (Written note from mom). My daughter has had past experience with Hour of Code and a few other coding games, but nothing like this (class). She was so excited and happy that she was able to design and create her own game instead of coding the movements of predetermined game layouts, which never held her attention long term.

Astonishing Variety of Stories


Its realy cold


AWE
SOME!

Programming Class

- At half-way point, each girl gives a demonstration.
- Take a break. Move.
  - Play "Programmer Says" game using conditional clauses. Each girl has a turn as programmer.
- After break, everyone follows with me as we animate the ballerina and add background music.
- Demonstrations again at the end.


ThinkPad
lenovo


here to be
AWESOME
DREAM
GIRL SCOUTS USA
OREGON AND SW WASHINGTON
5096
2016-


THE PERIODIC TABLE
OF ELEMENTS

Methods for increasing complexity

- Praise experimentation.
  - Some smart girls hate to make mistakes.
  - Some girls are timid and leave every variable at its default value
- Emphasize ownership. “You’re the programmer; you get to choose”
- Role-model optimization
- Teen demos
  - Increased complexity
  - Making a mistake (no embarrassment)

Memorable quotes

- "My friend, Mark, he's a boy, said that I couldn't learn SCRATCH, but I just did".
- "I didn't really want to come, but my friend Dana came, so I did too. Can I sign up for the next class?"
- (from a mom) "I had NO IDEA she would like something like this."

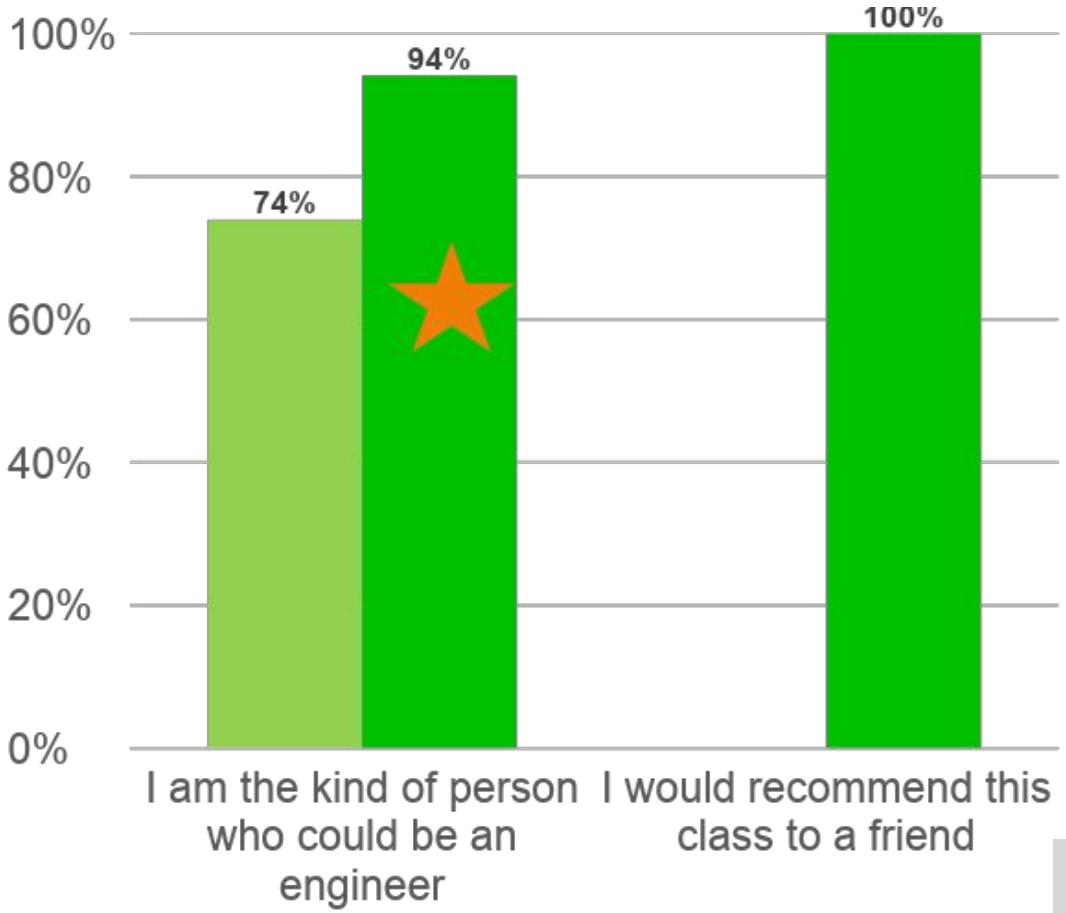
New in 2020: Robots

Adults needed

To build confidence and competence

Robots increase Self identification as Engineers

Robots class 2020



We ask this question only after the class

What is impact?

The goal is that every girl is successful at learning new skills and that she has CONFIDENCE based on COMPETENCE to take the next step in engineering learning.

GET Result

Girls Engage Technology

1903 Girls Served

2015-2021

SA Result

2019:

- OE STEM classes 318 more boys than girls
- GET 357 girls
- TOTAL: ~gender parity


Thank You
MacBook Pro
digital Story: Goals

Back-up

Teaching programming to young students using Scratch

Project	Goals	Concepts/ Vocabulary	comments
Create a Digitized Story 3-6th grade	Write the code to produce a unique story. Include · at least 3 characters, each with a program that contains movement, and conversation, · loop(s), · sound(s) and · backdrop(s).	· Sequence of code · Starting Conditions · Variables · Loops · Conditional Clauses (IF THEN) · x,y positions of characters At break, we play the Programmer Says Game, where each student takes a turn as the programmer and gives (a few) Conditional Clauses commands e.g. "programmer says, IF you are wearing blue, THEN do 3 jumping jacks"	Great project for beginners because there are multiple paths to success. Story length & complexity is variable. For all girl classes, refer to every non-gendered character as SHE. Engineers say "it doesn't work YET." Each student demonstrates their project to the group at least twice during a 3-hr session. (exceptions made if there is strong resistance).
	Advanced: implement animation. Sound editing.	How is motion different than Animation?	
	Advanced: importing images legally from internet (or images of their pets)	Usage Rights Creative Commons license	
Animated Birthday Card 3-4th grade	Choose a person to create a birthday card for. What characters would they like? Will there be cake and dancing? Ballons? Happy Birthday song?	Re-inforcement for all beginning concepts	Good for "the littles". Short project if you need a filler. No new concepts.
Animated Zoo 3-4th graders	ANIMATE at least three characters in your zoo. May draw your own background. Use animal sounds.	Interaction with the player (when this sprite clicked) Starting conditions Animation	
5th-7th graders	Advanced: create your own costumes for existing or drawn character. Record sounds		
	Older students could create an animated dance party with background music		
Digital Game Design 4-8th graders	Create a digital game that · is interactive · has clear goal(s) · has scores or rewards · is fun	· Player interaction · Sensing commands · Forever loops · Creating variables	Teach several methods for interaction: arrows to move, "when touching color/character/mouse", asking questions
	Advanced:	· Subroutines · Multiple levels	