# Adapting the Curriculum for Student Success
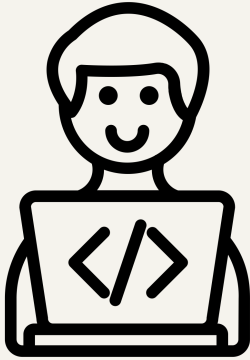
GE Academy

Robbie Snellman
Math

Daniel Zappala
Computer Science

# Creation of Math 109
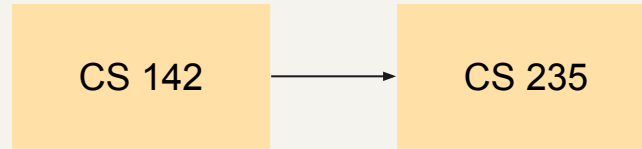
# Creation of CS 110
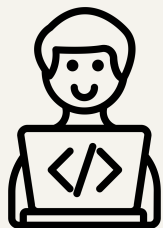
# For many years…

People who know how to code

CS Major + some STEM majors
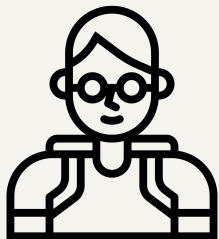
CS 142 → CS 235

# Now lots of people want to code
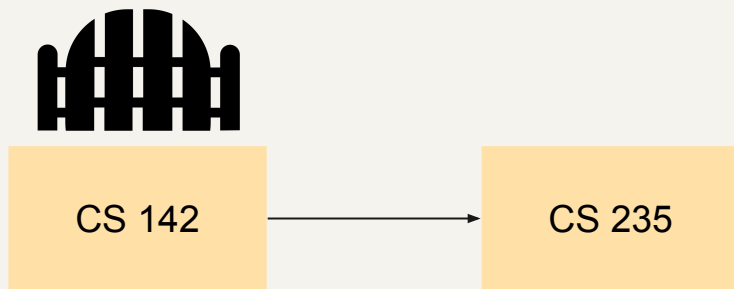
People who know how to code

Scientists + Engineers

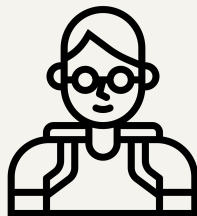Students from all majors

CS 142 → CS 235
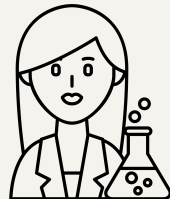
# Option 1 – gatekeeping



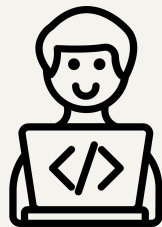| CS 142 | → | CS 235 |

- Instructors expect prior experience
- Students view the course as a "weeder" course
- Many leave for other majors (IT, IS)
- Many skip computing altogether
- Intro courses proliferate in other departments
- **~10% CS majors are women**
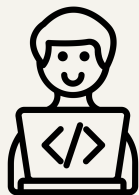
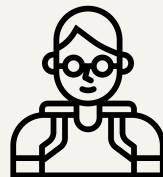# Option 2 – teach how to program

CS 142 → CS 235

- Instructors expect NO prior experience
- Students with experience are bored and make less experienced students feel out of place
- Students not well prepared for CS 235
- Faculty complaining that our students don't have strong fundamental skills in subsequent courses

# CS 110 + CS 111

High school, AP CS, online courses, jobs

CS 110 → CS 111 → CS 235

- CS 110 teaches beginners how to program
- CS 110 serves other programs on campus
  - *90%+ are not CS majors – some become CS majors*
  - *Mechanical Engineering sending 200+ students, more may come*
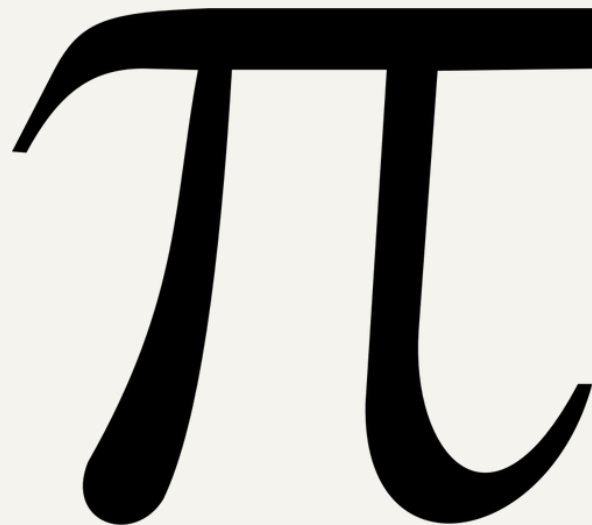- CS 111 provides strong fundamentals for students who proceed in CS or need it in their major

**01**

# How do you build a class designed for all students to succeed?

# Math 109

- Gather data on student needs
- Try something new
- Build the course for student success

π

# CS 110

- Graphical programming for the first month
- Go slow with concepts
- Low stakes practice
- Just in time teaching



Fix the Bridge

# CS 110



How would you rate your ability to program...

# Notes

- TBD

## 02

How can you help students feel like they belong in your class?

# Math 109

- The classroom can be a community
- Group work during class
- Formative feedback in safe and open environments

# CS 110

- Labs and lab partners
  - *Learn to program together*
  - Small ~20 - 30 students (lectures are ~200)
  - Can look at each other's code, write code together, and talk about different ways of solving problems
- CSIDE mentors
  - Peer mentors, mostly women and international students

How can you help students feel like they belong in your class?
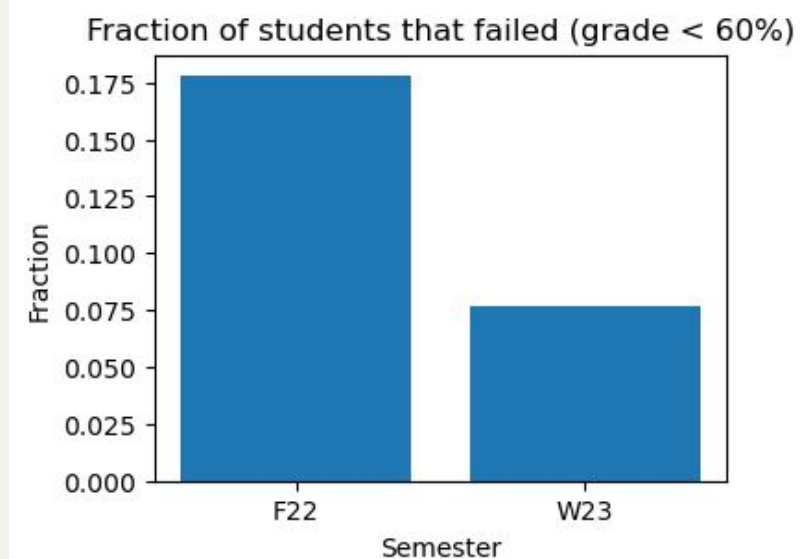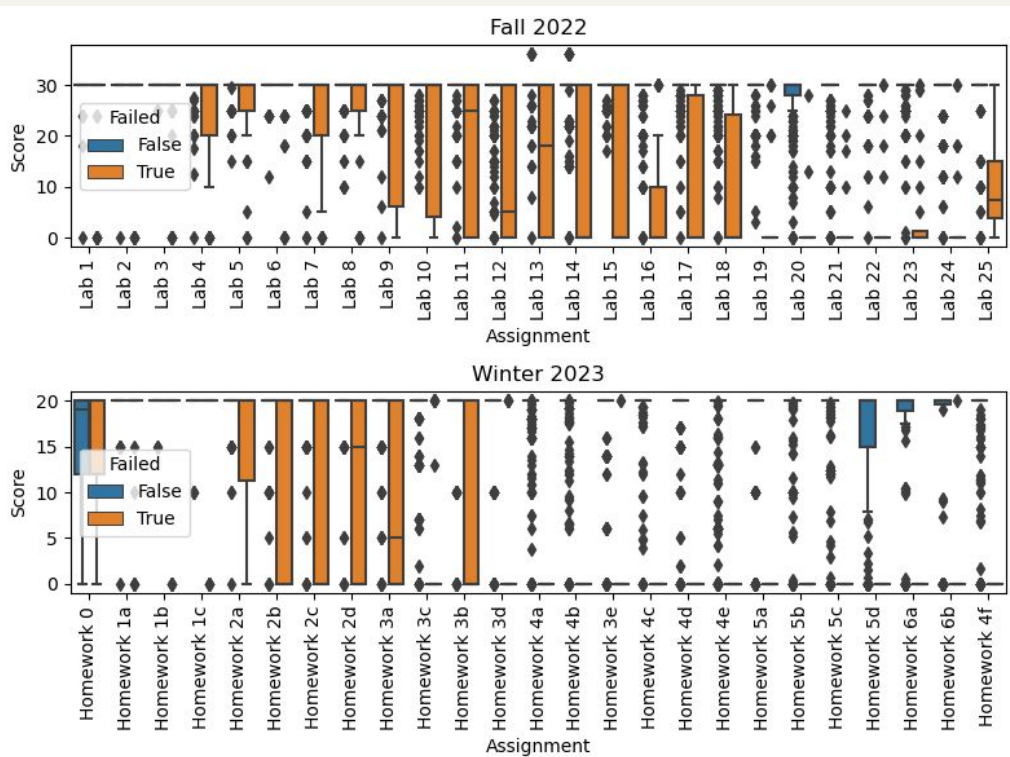
# Notes

- TBD

**03**

# How can you measure student performance and adjust over time?

# Math 109

- Gather data from the students directly (e.g. surveys, midcourse evals, SCOT program, etc.)
- Modify the curriculum accordingly
- Understand what you are measuring

# CS 110

# Notes

- TBD

# 04

How do you build assessments that test competencies without built in assumptions?

# Math 109

- Be intentional regarding what students need to know, not what we think they should already know
- Introduce benchmark quizzes early on to address any knowledge gaps
- If trends are apparent in the quizzes, then modify the curriculum to introduce sooner those concepts related to knowledge gaps
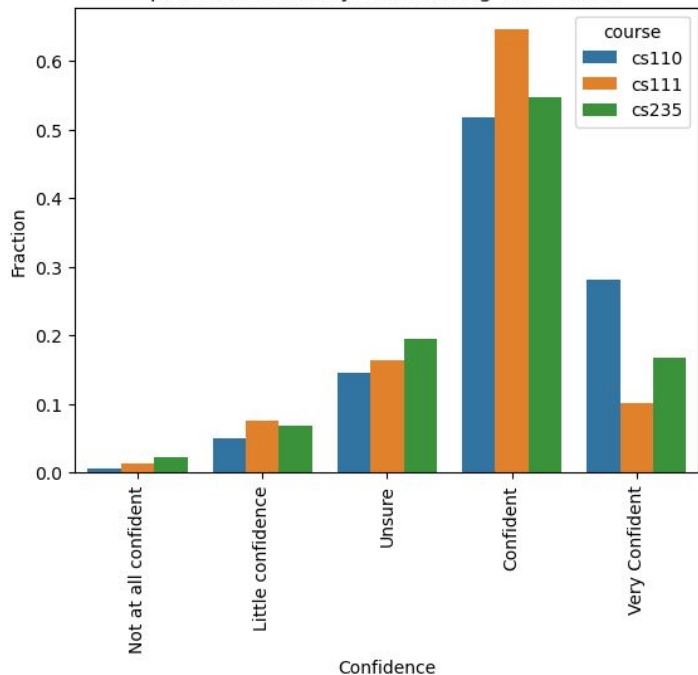
# CS 110

- Learning outcome: Write code from scratch to solve a problem.
- After first semester, manys students couldn't do this
  - Faulty assumption: "Training wheels" most of the semester → a few weeks without training wheels would be enough
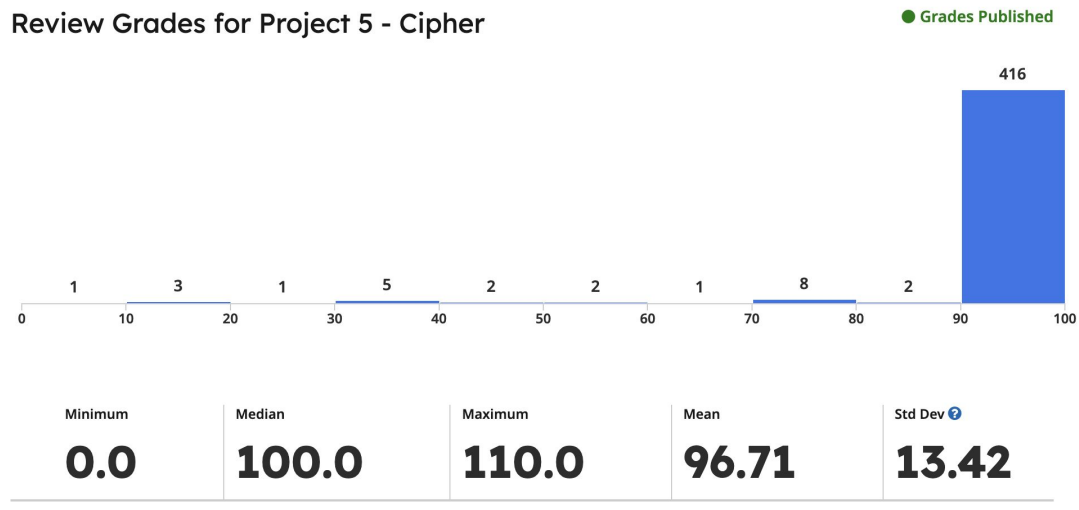- Redid the class to teach this skill early and provide lots of practice

# How do you build assessments that test competencies without built in assumptions?

# CS 110



How confident do you feel in your personal ability to independently write code to solve similar problems to what you did during this course?



Review Grades for Project 5 - Cipher    ● Grades Published

| Minimum | Median | Maximum | Mean | Std Dev |
|---|---|---|---|---|
| 0.0 | 100.0 | 110.0 | 96.71 | 13.42 |

# Notes

- TBD

# Thanks!

Do you have any questions?

snellman@mathematics.byu.edu
zappala@cs.byu.edu