The Many Futures of CS Education

SIGCSE *L.A.* Panel+Discussion
February 18, 2026

Panelist introductions...

Lauren Bricker
UW

here!

Julie Medero
HMC

here!

Adam Blank
Caltech

arriving?

Zach Dodds
HMC

here!

Radical Inclusivity

Lauren Bricker, University of Washington
bricker@cs.washington.edu

What percentage of students who enter the college intending to study engineering actually complete engineering degrees?

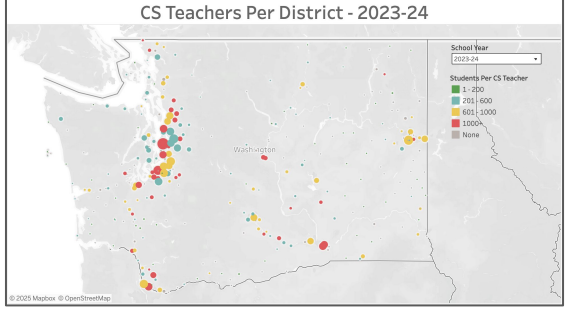
- Nationally 4 year graduation rate: 30%, 6 year: 55% (ASEE, 2017)
- Historically at UW, it's 50%

National statistic from https://ira.asee.org/wp-content/uploads/2017/07/2017-Engineering-by-the-Numbers-3.pdf
UW Statistist from https://www.engr.washington.edu/news/article/2017-09-18/engineering-student-success-stars

Barriers to entry

Historically **25%** of first-generation, low-income students, and students from underserved communities who enter the UW as pre-engineers will earn a degree in the field

These students also have difficulty getting admitted to engineering programs because of inadequate HS preparation



Your ZIP code shouldn't determine whether you can become an engineer.

– Dr. Eve Riskin Former Associate Dean for Diversity and Access College of Engineering

We can't expect students to be college-ready; we need to be student-ready colleges.

- Dan Greenstein, formerly of the Gates Foundation and Pennsylvania Chancellor on a panel in September 2019 that Dr. Riskin attended

https://www.engr.washington.edu/news/article/2017-09-18/engineering-student-success-stars
https://www.cstawa.org/2025/08/25/demo-washington-cs-teacher-population-2019-24/

Pathways for Inclusive Excellence

HS Senior Year
PIE Pathway Admissions

Students are identified during the university admission process and assigned STARS (2 years) or Allen Scholars (1 year)

Year 1
First year support

Academic support, cohort building, near peer mentor support, and proactive advising

Program completion
Conditional Admission

Transition to their major upon completion of the program requirements

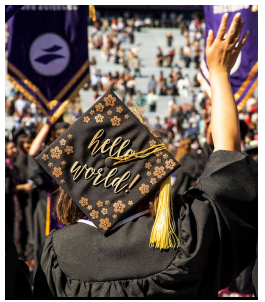
Pre-Autumn
Summer Bridge Course

Four week summer transition program

Year 2
Second year support (STARS)

Continued academic support, cohort building, near peer mentor support, and proactive advising

UW Graduation



CSE supplemental workshop course structure

- **Workshops** (once weekly) to reinforce and practice concepts.
  - Active learning, collaborative group work on worksheets and whiteboards
  - Twice quarterly small group presentation problems with TA feedback*
  - Mock midterm and final with TA lead Show What You Know (SWYK)*
  - Weekly assessments with quiz corrections if needed
- **50-minute work sessions**
  - **25-minute check-in meetings** (once weekly) with TAs
  - **25-minute study time** with post work reflection
- **Weekly pre-work** due before workshop and check-in meetings.

* Oral assessments of knowledge

Recent program results

STARS results (cohorts 1-11 (2013-2023))

- 352 students served
- 96% retained in year 1, 82% retained through year 2
- 78% graduate

Detailed data from students Fall 2015-Fall 2017

Time to Graduation	4 years	5 years	6 years
UW Seattle grad from UW Seattle	68.4%	80.8%	82.3%
STARS grad from UW Seattle		82.7%	85.7%

Allen Scholars results (from Fall 2022)*

- 156 students served
- 88% are first-generation college students and 65% are low-income students compared to Allen School demographics (26% and 18% respectively).
- > 96% of Allen Scholars students have successfully completed the Allen Scholars program and transitioned into an Allen School major

* Allen Scholars changed to a full conditional admit program in 2022. Previous data not tracked.

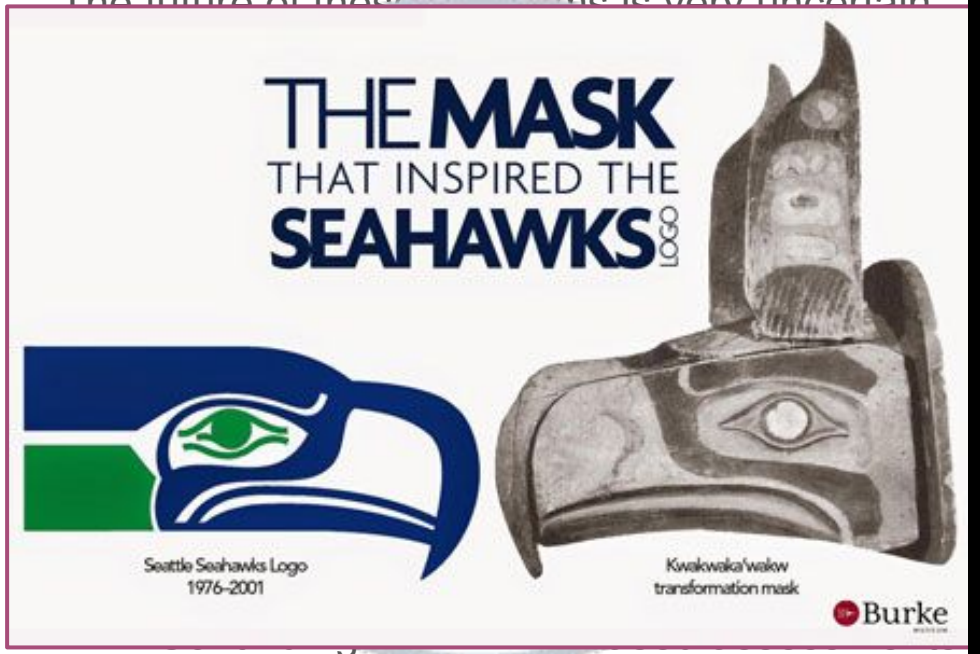
The North Star…

Students don’t have the added cognitive overhead of navigating support at UW…

…the support is built in to their program

The future…

The future of these [illegible]ams is very uncertain:



corrections) to engage students in the learn[illegible]

Los Angeles isn't the only "**LA**" ...

Human-in-the-Loop CS Education

Julie Medero, HMC
medero@cs.hmc.edu

My **flavor** of Intro CS:

- Part of a pass/fail gen ed requirement for all first year students
- Aimed at students with ~one year of high school CS (e.g, AP CSA)

Course Goals:

- Eliminate the “**magic**” from CS (transistors through proofs of uncomputability!)
- Expose students to the breadth of CS
- Manage *wildly* different prior experience, without any extra “credit” as a carrot

Human-in-the-Loop CS Education for me:

Content in class varies each year based on student self-assessments

Human-in-the-Loop CS Education for me:

Flexible paths through assignments, with recommendations based on student self-assessments

Print Debugging in Python

Print Debugging in Python

Print Debugging in Python

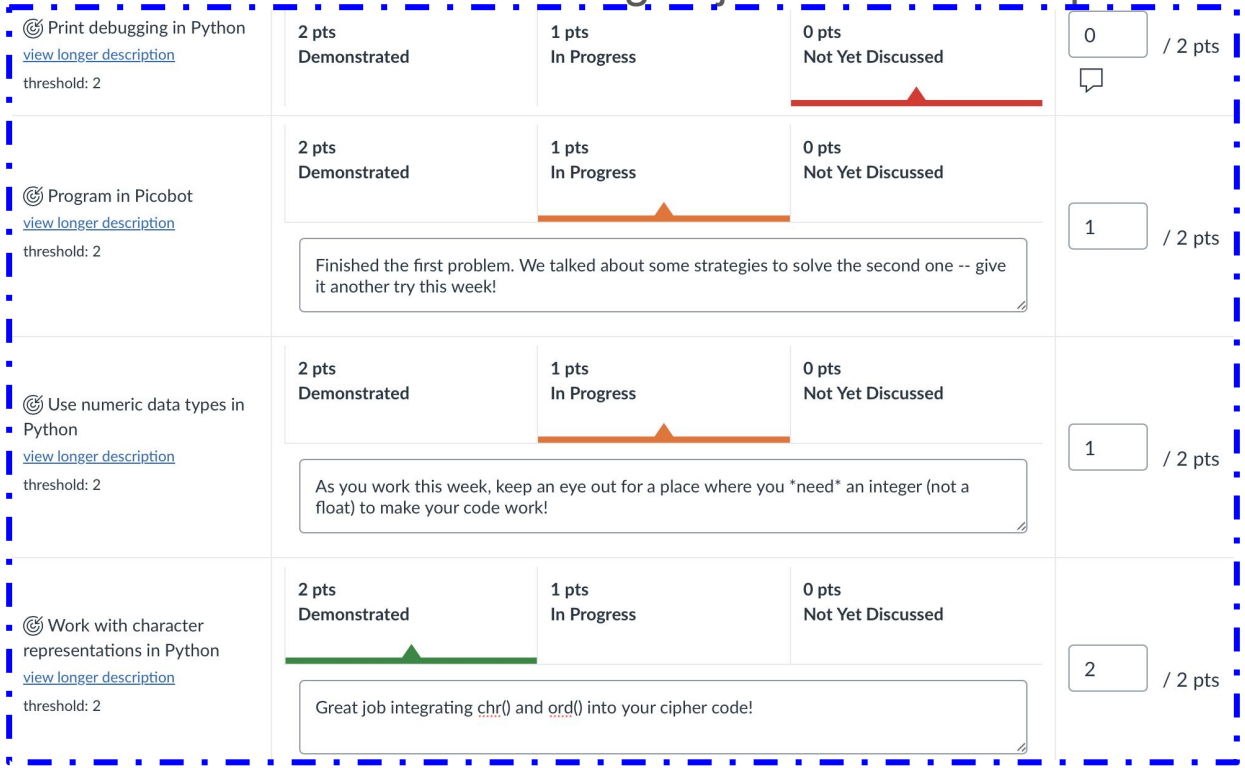
Overview: Strategically use print statements to debug Python functions.

Relevant CS 5 Gold Homework Problems: N/A. To demonstrate this learning objective, you should document one Python bug that you found with a print statement. Be prepared to explain to your prof (1) how you decided to place the print statement and (2) what it revealed.

Challenge Options: Consider exploring the python logging library for another way to track messages from your code. What are the advantages of using a logger? Why might you still sometimes want to use print statements?

Human-in-the-Loop CS Education for me:

Standards-based grading: students can demonstrate learning objectives *any* week, but need to demonstrate *all* learning objectives at basic proficiency to pass



Human-in-the-Loop CS Education for me:

No grading of code/artifacts: All grading happens in one-on-one meetings with the instructor

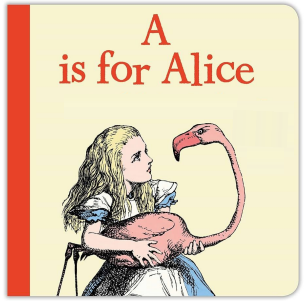
Human-in-the-Loop CS Education for me:

“High pass” grades are based on *experiences*, not just assignments.

What's working (better)?

- Artifacts both over- and under-estimate student understanding
- Conversations focus on *understanding*, which removes the incentive to create an artifact at an arbitrary deadline
- Flexible deadlines let students go back for more practice
- Student-informed content with buy-in
- Flexibility helps build a cohort identity
- Students learn lessons (e.g., about the importance of documenting code) authentically, not as side-effects of assignment requirements

What's … *not*?



CS is for *Community Shaping*

Zach Dodds, HMC
dodds@g.hmc.edu

What *should* Higher Ed do?

Explore techniques for students' self-expression, proactive learning, and societal agency.

- *Computing* is valuable on all three fronts...
- *Conversation* is valuable on all three fronts...

We're *universally* using *Computing Conversations* Fall25+Spr26

- Q'n 1: Can we scale Julie's one-on-one conversations using TAs?
- Q'n 2: Can we scaffold them so that *all participants* emerge happier?

- Answer: We can at least *try*...

secret ingredient ~ *grutors*

i.e., a special blend of molecules known as "*people*"

Weekly Share-outs

195 students/week ~ 15 minutes/demo

Tuesday, September 30											
1:15-3:45pm session											
1:15-1:45pm	elia Tripp Wiseha	Charlotte Wong	Bethany Prado	Stella Gajar	Carol Wang	Jack Paavola	Nayah Torres	Claire Johnston	Maya Sprosts	Oliver Tumolo	ten spots in this half-hour
2:00-2:30pm	Ryan Bram	Janety Zhang	Jaahnavi	Peter Phan	Zara Ansari	Ethan Hoen	Oliver Kiehl	Harry Webster	Olivia Markowich	Aaron Gonzalez	ten spots in this half-hour
2:45-3:15pm	Cole Boake	Anya Sheng	William Jones	Charlie Hill	Raphael e	Alice Kelly	Cory Welch	Karen Kitchen	Beckett Heywood	Hayat	ten spots in this half-hour
3:15-3:45pm	Natalie Huang	Serafina Figlozzi	Cooper Giles	Michael Yao	Cindy Nguyen	<Jordan Yates>	<Terry W>	Wes	<>	Avi Palomino	ten spots in this half-hour
6:00-8:00pm session											
6:00-6:30pm	Emmie Ding	Angel Wu	Adriana D.	ryan yoo	Hanwen Miao		<name>	<name>	<name>	<name>	ten spots in this half-hour
6:45-7:15pm	Eileen Wang	Yichen Yao	izabeth Hernand	Zoe Craft	<Alexandra Ye>	<name>	<name>	<name>	<name>	<name>	ten spots in this half-hour
7:30-8:00pm	Brennan Tobin	Oliver Flor	Blaze Hedani	lackson Giumarra	evelyn law	cindy fan	Abdullah Ali			<name>	ten spots in this half-hour
8:00-10:00pm session											
8:00-8:30pm			Sihan Qiao	<name>	<name>	<name>	<name>	<name>	<name>	<name>	ten spots in this half-hour
8:30-9:00pm	Cassidy Goya	<name>	<name>	<name>	<name>	<name>	<name>	<name>	<name>	<name>	ten spots in this half-hour
9:00-9:30pm	<name>	<name>	<name>								ten spots in this half-hour
9:30-10:00pm	<name>	<name>	<name>								

Spreadsheet sign-up: 12 hours, 24 peer tutors

Wednesday, October 1											
1:15-3:45pm session											
1:15-1:45pm	Grace Regier	Jack Sample	Cate Lee	ramis Thai-Garic	Evan Davis	Caitlin Kuhlmann	Rain Hannsz	vetha Madhanraj	Jonah Mayo	Duc Nguyen	ten spots in this half-hour
2:00-2:30pm	Ysabel Chu	a Hernandez-Alv	Lucien O'Neil	Erin Weir	Richard Burrow	Star Grote	Gwen Severson	Matthias Kim	Chloe Polin	Vern Van Asdale	ten spots in this half-hour
2:45-3:15pm	Taylor Rienhart	William Ngo	Damian Argueta	Alan Liu-Sui	Ash Ragins	Ethan Chin	Aidan Cho	Erick Velarde	Stephanie Corre	Adriana L.	ten spots in this half-hour
3:15-3:45pm	Oscar Zhao	Amanda Phillips	Rain Salkever	Charlie Medina	abu Heikaus Wea	Gulzar A	<name>	<name>	<name>	<name>	ten spots in this half-hour
6:00-8:00pm session											
6:00-6:30pm	Abbie H	ddison Mae Mye	Anna Berman	Jairo Jasmin		<Avery Tin>	Elaina Garg	Olivia Yan	Kelly Zhou	Juntong Zhou	ten spots in this half-hour
6:45-7:15pm	Jorge Pacheco	Caleb Settles	Ethan Kim	Shawn Dahiyat	Tyler Chelew	Amelia Carlino	Diego Amescua	Sophia Rico	<name>	<name>	ten spots in this half-hour
7:30-8:00pm	Meha Chaudhary	Nastia Derepa	Kylie Brunelli		Jiselle Nguyen	Ivan Rugamba	Samuel Wang	Zoe Craft	<Joshua Rios>	<name>	ten spots in this half-hour
8:00-10:00pm session											
8:00-8:30pm	Outreach program spots (outside the 5Cs)										ten spots in this half-hour
8:30-9:00pm	Lily Horn	Molly McLaughlin	Abigail Stewart	Lawrence Notar	Arman Rudar	Shifali Dinesh	Luci Dunn	Michelle Lu	Sean Beltran	Katja Gentleman	ten spots in this half-hour
9:00-9:30pm	isabella rhee	misty bera	Frander Pena	Bella Jiao	Juliuz Salazar	Nia Ho	Matthew Cabrera	Leonardo Castro	Kylie Brunelli	<name>	ten spots in this half-hour
9:30-10:00pm	<name>	<name>	<name>	<name>	<name>	<name>	<name>	<name>	<name>	<name>	ten spots in this half-hour

Weekly Share-outs

195 students/week ~ 15 minutes/demo

Students set-up-shop...
Peer-tutors *rotate*...
~5 min each...

Weekly Share-outs

Each share-out "script" is known ahead of time...

hw3: *Demos*

T 9/23 and W 9/24

Name: _____ first + last

1. Ready? Grab one of these pages + set up your files...
2. Grutors: Circulate, keeping things fair + flowing
3. ***Each demo***: Aim for 2-5 minutes...
4. (If it's busy, we can use the board to keep track)
   4.1 Pairs can demo "individually, together"!
   4.2 ***Stickers*** optional... Check out when finished...

sound

grutor

1. Show off your sound file...
   Q'n: Do you have **reverse**? **static**? **echo**? If so, demo!

   Q'n: [Entirely Optional] **chord**? Something else?
   If so, show off these!

2. [Thought experiment] Would speech-recognition software have an easier or harder time learning to ***interpret (a) reversed speech, (b) static speech, (c) echoing speech***, as compared to humans? Is the answer (easier/harder) the same for all three, or different from case to case?

strings

grutor

1. In VSCode, show off your `decipher` function:
   (a) Try it at the terminal, on a string of your choice where it works.
   (b) Make up a real English sentence where it fails! (Hint: use text for its own sake, e.g., "Zqvq isn't a real word!" or "It sounded like Zzzz..." Demo your own variation...
   (c) Why does the code not "work" in this latter demo?

2. In VSCode, show off your `jscore` function:
   (a) Show two 5-letter words that get a 0.
   (b) Show two 5-letter words that get a 5, but aren't the same!
   (c) Would it be easy or hard to change `jscore` so that it only scored letters at the same index? (Wordle-green vs. Wordle-gold)

the BFF

"the blank-file function"

Demo a call to a new API – and extract a piece of data from it!
(*) Check out the website **`pokeapi.co`**
(a) Try their "direct link to results" API call, for **ditto**:
(b) Using copy-and-paste in a new cell in your hw3pr2 API Colab notebook, make the same call *programmatically* from Python...
(c) show that you can obtain the same results in the notebook as in the browser directly
(d) from the resulting dictionary (json), deliberately extract one piece of data ... all the way down to a string or a number
(e) Run the same code on a ***different*** pokemon!

grutor

hw5: *Demos*

T 10/7 and W 10/8

Name: _____ first + last

1. Ready? Grab one of these pages + set up your files...
2. Grutors: Circulate, keeping things fair + flowing
3. ***Each demo***: Aim for 2-5 minutes...
4. (If it's busy, we can use the board to keep track)
   4.1 Pairs can demo "individually, together"!
   4.2 ***Stickers*** optional... Check out when finished...

add!

grutor

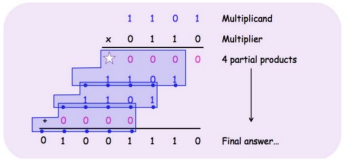
A. Show off your Ripple-Carry Adder...
   (1a) Here, choose four bits for the top input: y3,y2,y1,y0
   What decimal value is this?
   (1b) Next, choose four bits for the other input: x3,x2,x1,x0
   What decimal value is this?
   (2a) Explain what the decimal value is for the result (the sum)
   (2b) Did it work?!

B. Delete one Full Adder **row** -- just disconnect one AND's inputs
   (1) Note which input bits are now incorrect ...
   (2) Then, create a ripple-carry example that is *also* incorrect ...
   (3) Explain how the error shows up in the ripple-carry!
   Finally, be sure to fix your Full Adder

multiply!

grutor

1. Show off your multiplier
   (a) Does 6x7 work? Does 7x6 work? 15x15? Others?
   (b) Show where some of the digits/dots in this picture of the multiplication algorithm match those in your circuit:



the BFF

"the blank-file ~~function~~ circuit"

Head to this url:

`circuitverse.org/users/26206/projects/bff-but-for-circuits`
(1) Create a new fork – or copy – of this public circuit.
(2) It has a truth table with four input bits and one output bit.
(3) What are the **values** of the inputs that have 1 as output?
(4) Build this circuit! You can use "standard minterm" ... OR ...

Optionally ... you can try to optimize with fewer AND gates!
We don't know the minimum ... yet

extra Divider?

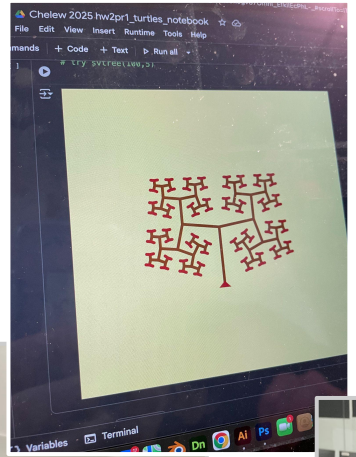
grutor

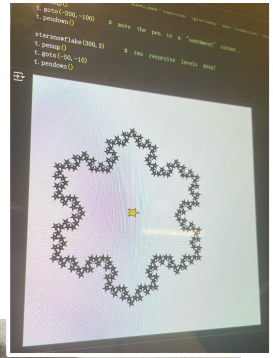
Three components of ~5 minutes each. The "BFF" is the *Blank-File Function*

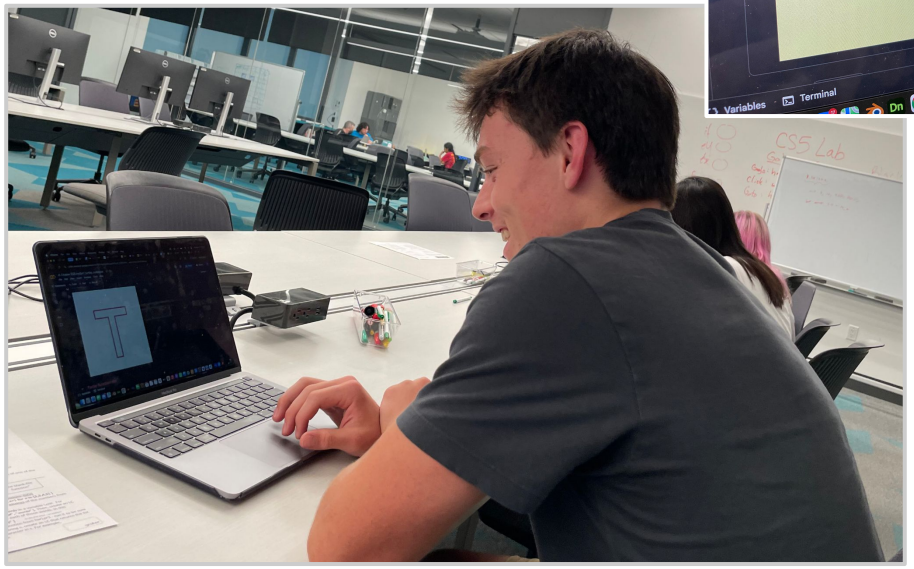
Weekly Share-outs

Weekly Share-ins

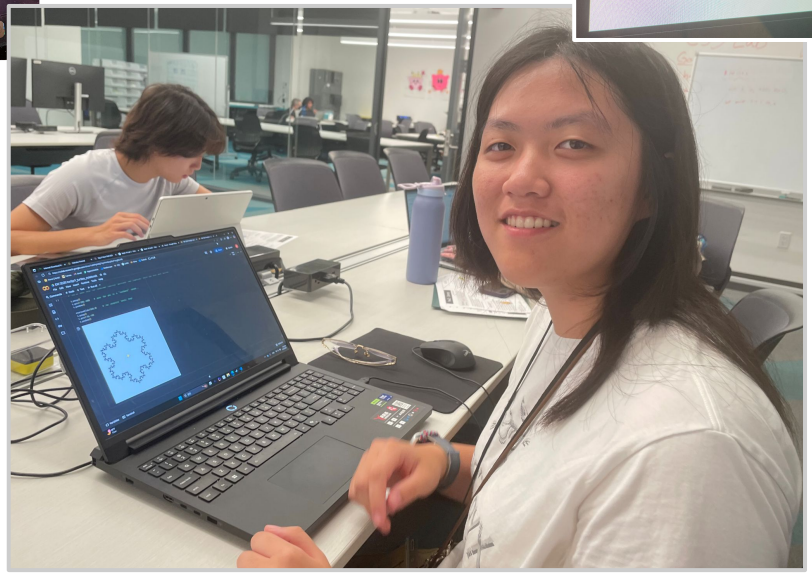
Celebration of creative construction







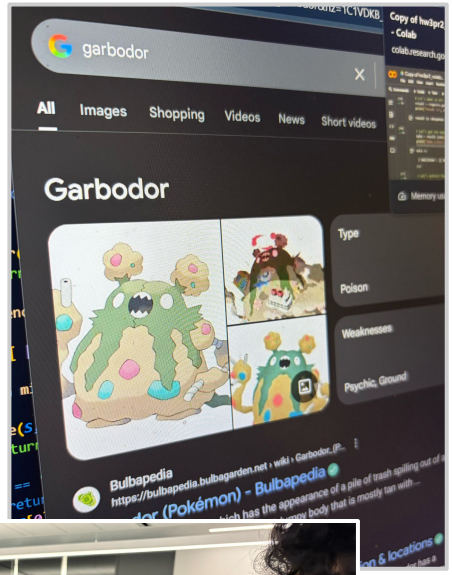
Tyler



Eileen

Weekly Share-outs

Weekly Share-ins



the BFF "the blank-file function"

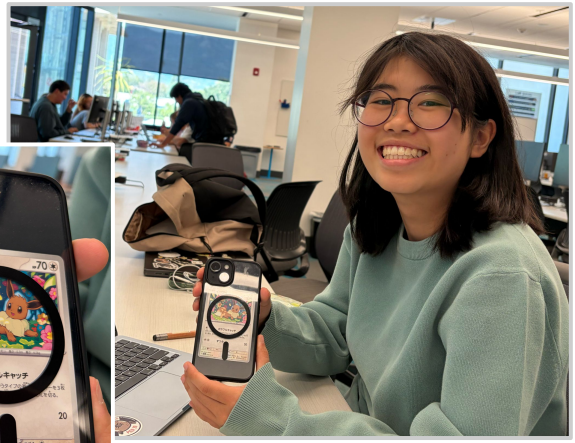
Demo a call to a new API – and extract a piece of data from it!

- (*) Check out the website **pokeapi.co**
- (a) Try their "direct link to results" API call, for **ditto**:
- (b) Using copy-and-paste in a new cell in your hw3pr2 API Colab notebook, make the same call *programmatically* from Python...
- (c) show that you can obtain the same results in the notebook as in the browser directly
- (d) from the resulting dictionary (json), deliberately extract one piece of data ... all the way down to a string or a number
- (e) Run the same code on a ***different*** pokemon!

grutor

Charlie

Sean



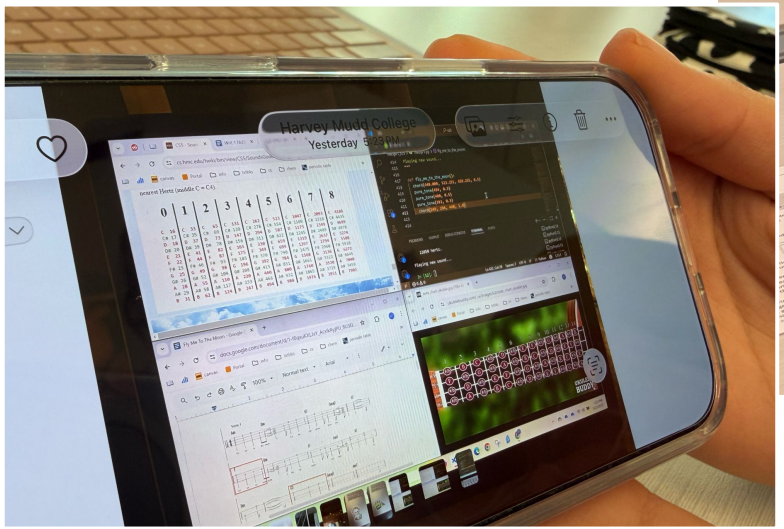
Charlotte

Celebration of shared interests ...

Weekly Share-outs

Weekly Share-ins

Workflow Insights!



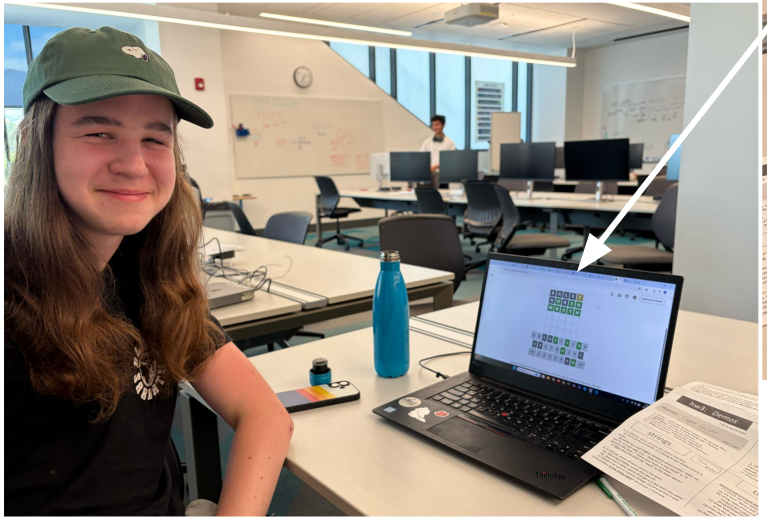
Naya and Cooper

Not Complete? Not complete *yet* ...

Weekly Share-outs

Weekly Share-ins

Sentimentflow Insights!



Cooper

Naya and Cooper

Not Complete? Not complete *yet* ...

The Many Futures of CS Education

SIGCSE *L.A.* Panel+Discussion
February 18, 2026

The One Future of CS Education

Computing centers the interpersonal …

… *however challenging* that might be!

SIGCSE *L.A.* Panel+Discussion
February 18, 2026

Provocations + Prompts:

What is being taught now that won't be relevant in the future?
What do you feel you're "forced" to include? An, *by whom*?

What skills/topics will be more important in the future than now?

What do we value now that will have lasting value *well beyond* what is being acknowledged by this moment's headline cycle?

Is the audience for computing skills and sophistication growing? *Should it be?*

What should students know *experientially* about
(a) LLMs + AI? (b) Computing? (c) College+Career? (d) Life?

What else should we be asking or talking about?!?

Debrief + Declaim...

Provocations + Prompts:

What is being taught now that won't be relevant in the future?
What do you feel you're "forced" to include? By whom?

What skills/topics will be more important in the future than now?

What do we value now that will have lasting value *well beyond* what's acknowledged by the current hype cycle?

Is the audience for computing skills and sophistication growing? *Should it be?*

What should students know *experientially* about
(a) LLMs + AI? (b) Computing? (c) College+Career? (d) Life?

What else should we be asking or talking about?!?

Thank you!

Provocations + Prompts:

What is being taught now that won't be relevant in the future?
What do you feel you're "forced" to include? By whom?

What skills/topics will be more important in the future than now?

What do we value now that will have lasting value *well beyond* what's acknowledged by the current hype cycle?

Is the audience for computing skills and sophistication growing? *Should it be?*

What should students know *experientially* about
(a) LLMs + AI? (b) Computing? (c) College+Career? (d) Life?

What else should we be asking or talking about?!?

Radical Inclusivity

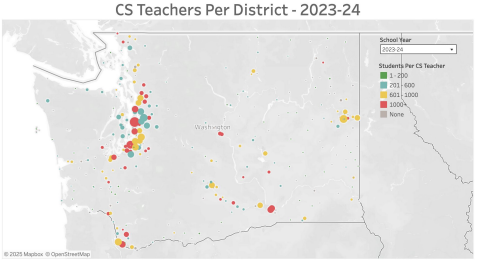
Lauren Bricker, University of Washington
bricker@cs.washington.edu

What percentage of students who enter the college intending to study engineering actually complete engineering degrees?

- Nationally 4 year graduation rate: 30%, 6 year: 55% (ASEE, 2017)
- Historically at UW, it's 50%

National statistic from https://ira.asee.org/wp-content/uploads/2017/07/2017-Engineering-by-the-Numbers-3.pdf
UW Statistist from https://www.engr.washington.edu/news/article/2017-09-18/engineering-student-success-stars

Barriers to entry



Historically **25%** of first-generation, low-income students, and students from underserved communities who enter the UW as pre-engineers will earn a degree in the field.

These students also have difficulty getting admitted to engineering programs because of inadequate HS preparation.

Your ZIP code shouldn't determine whether you can become an engineer.

– Dr. Eve Riskin Former Associate Dean for Diversity and Access College of Engineering

We can't expect students to be college-ready; we need to be student-ready colleges.

- Dan Greenstein, formerly of the Gates Foundation and Pennsylvania Chancellor on a panel in September 2019 that Dr. Riskin attended.

https://www.engr.washington.edu/news/article/2017-09-18/engineering-student-success-stars
https://www.cstawa.org/2025/08/25/demo-washington-cs-teacher-population-2019-24/

Pathways for Inclusive Excellence

Admissions

Students are identified during the university admission process and assigned to one of the following based on need:

- STARS Engineering (1 year)
- STARS Engineering or Allen School (2 years)
- Allen School Scholars Program (1 year)

First year support

- Supplemental workshops (Math, Science, CSE, (for Allen Scholars), etc)
- Near peer mentor support
- Required cohort meetups
- Intrusive advising (as needed)

Conditional Admission

Completion of the Program/requirements is a condition of their full admission to the department or school

- Commit to participate
- Culture of excellence & support
- Discomfort is essential for growth

HS Senior Year

Pre-Autumn

Summer Bridge Course

4 week summer transition program (must live on-campus)

- Cohort building
- Intensive STEM course materials (different for STARS and Allen Scholars)
- College Navigation

Year 1

Year 2

Second year support

- Required second year support for STARS students (when STARS takes CSE courses)
- Majority of Allen School students “graduate” the program. Those who don’t get continued support for one more quarter.

Program completion

UW Graduation

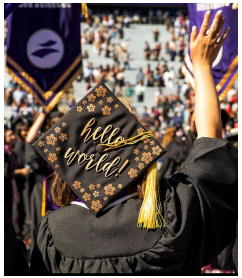


Image from: https://www.washington.edu/boundless/2019-in-focus/

CSE supplemental workshop course structure

- **Workshops** (once weekly) to reinforce and practice concepts.
  - Active learning, collaborative group work on worksheets and whiteboards
  - Twice quarterly small group presentation problems with TA feedback*
  - Mock midterm and final with TA lead Show What You Know (SWYK)*
  - Weekly assessments with quiz corrections if needed
- **50-minute work sessions**
  - **25-minute check-in meetings** (once weekly) with TAs
  - **25-minute study time** with post work reflection
- **Weekly pre-work** due before workshop and check-in meetings.

* Oral assessments of knowledge

Recent program results

STARS results (cohorts 1-11 (2013-2023))

- 352 students served
- 96% retained in year 1, 82% retained through year 2
- 78% graduate

Detailed data from students Fall 2015-Fall 2017

Time to Graduation	4 years	5 years	6 years
UW Seattle grad from UW Seattle	68.4%	80.8%	82.3%
STARS grad from UW Seattle		82.7%	85.7%

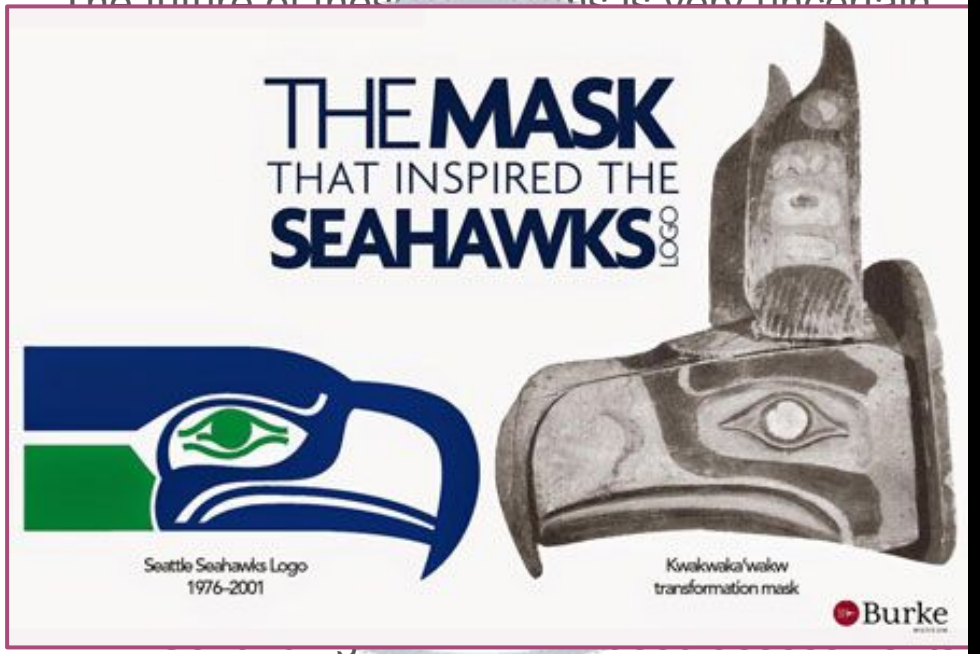
Allen Scholars results (from Fall 2022)*

- 156 students served
- 88% are first-generation college students and 65% are low-income students compared to Allen School demographics (26% and 18% respectively).
- > 96% of Allen Scholars students have successfully completed the Allen Scholars program and transitioned into an Allen School major

* Allen Scholars changed to a full conditional admit program in 2022. Previous data not tracked.

The future…

The future of these [illegible]ams is very uncertain:



corrections) to engage students in the learn[illegible]

Los Angeles isn't the only "**LA**" ...

What's … *not*?

Spending our time where it counts!

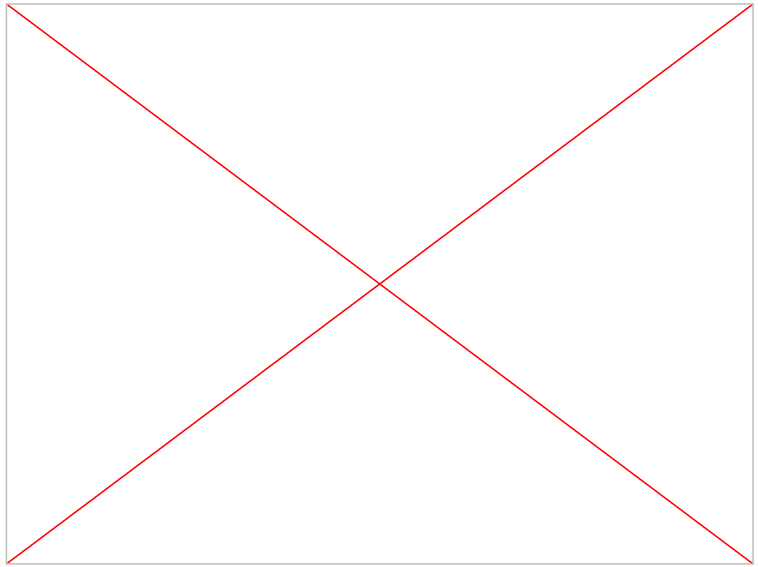
Adam Blank, Caltech
blank@caltech.edu

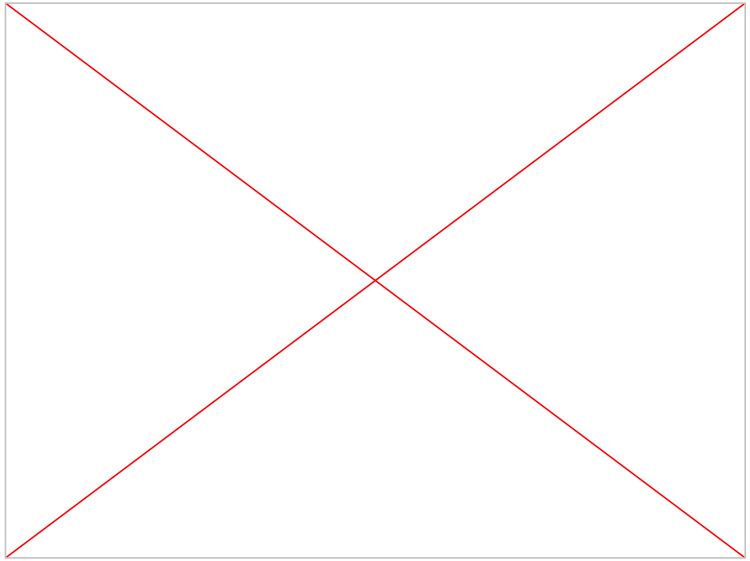
Courses are *ecosystems…*

- **Actors**: students, TAs, instructors, staff
- **Products**: grades, code artifacts, learning (???)

What products are important enough to spend actor time on?

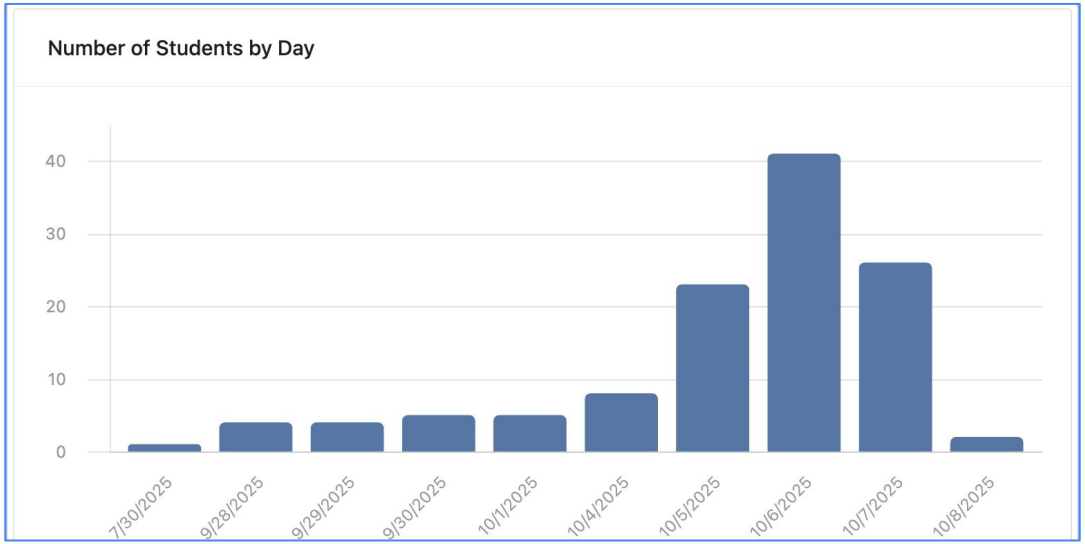
What products are important enough to spend actor time on?

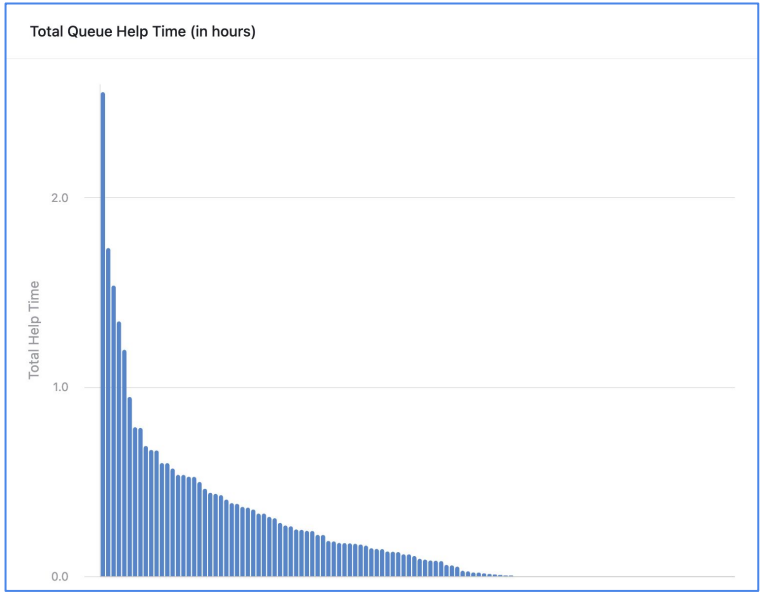




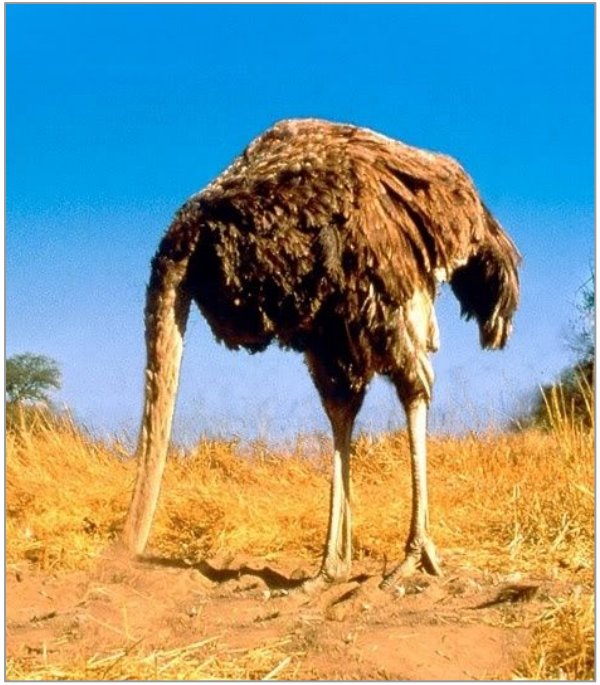
What products are important enough to spend actor time on?

What products are important enough to spend actor time on?





Before:

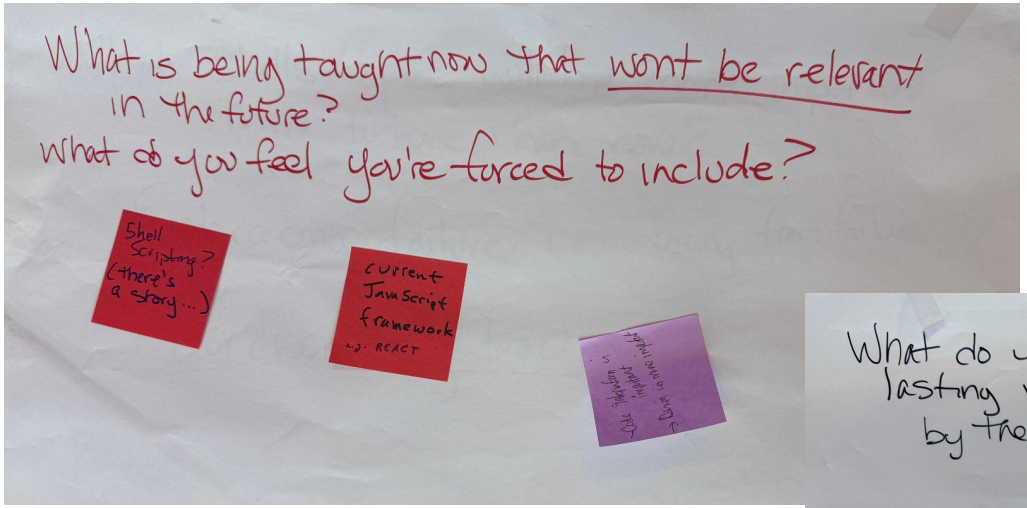


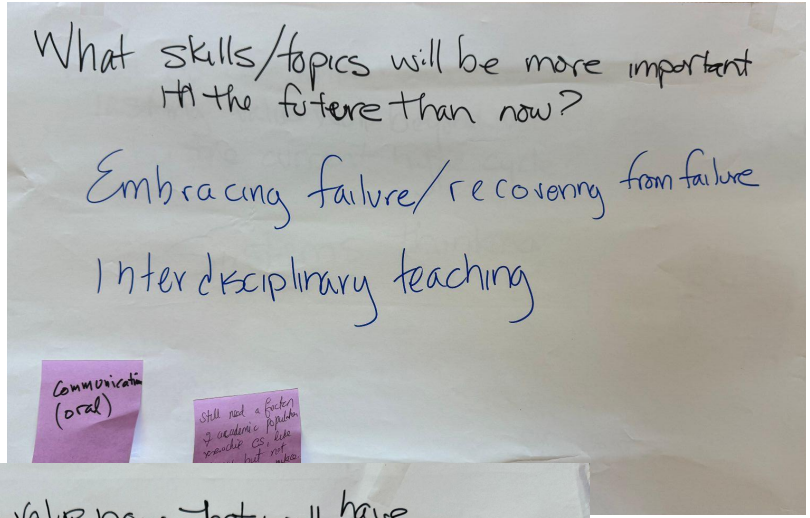
After:

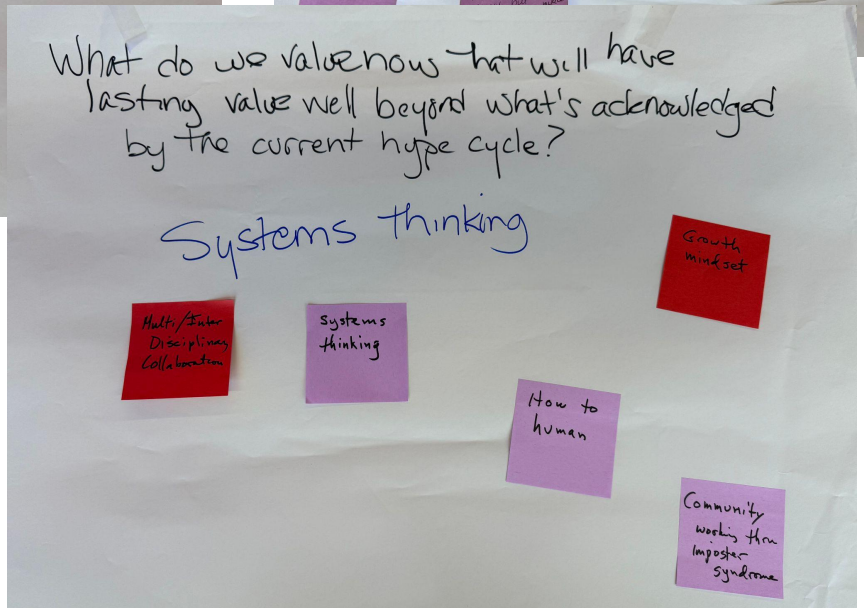

kül
SKINNY POP POPCORN
10 MILLION LUNCHES
Lay's
Doritos
RUFFLES
Fritos
Cheetos
classic
18 BAGS
STIRRER
organic
honeycrisp apples

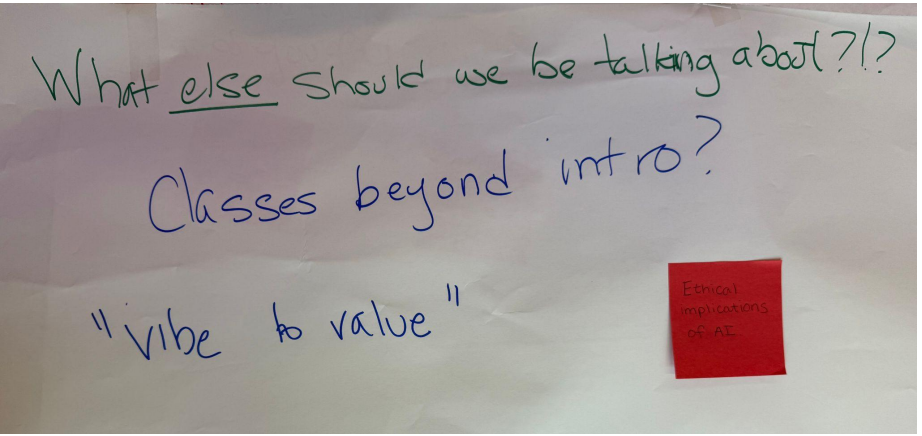

EXIT
CALTECH

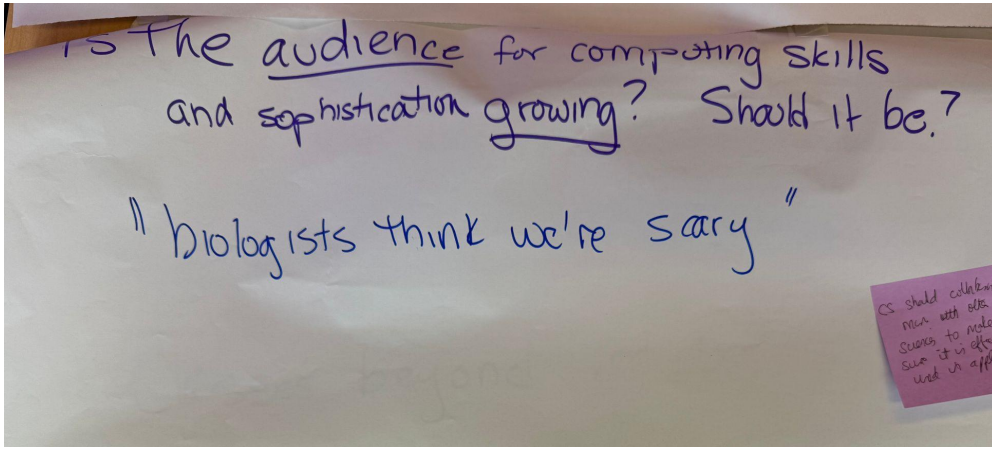
Share out

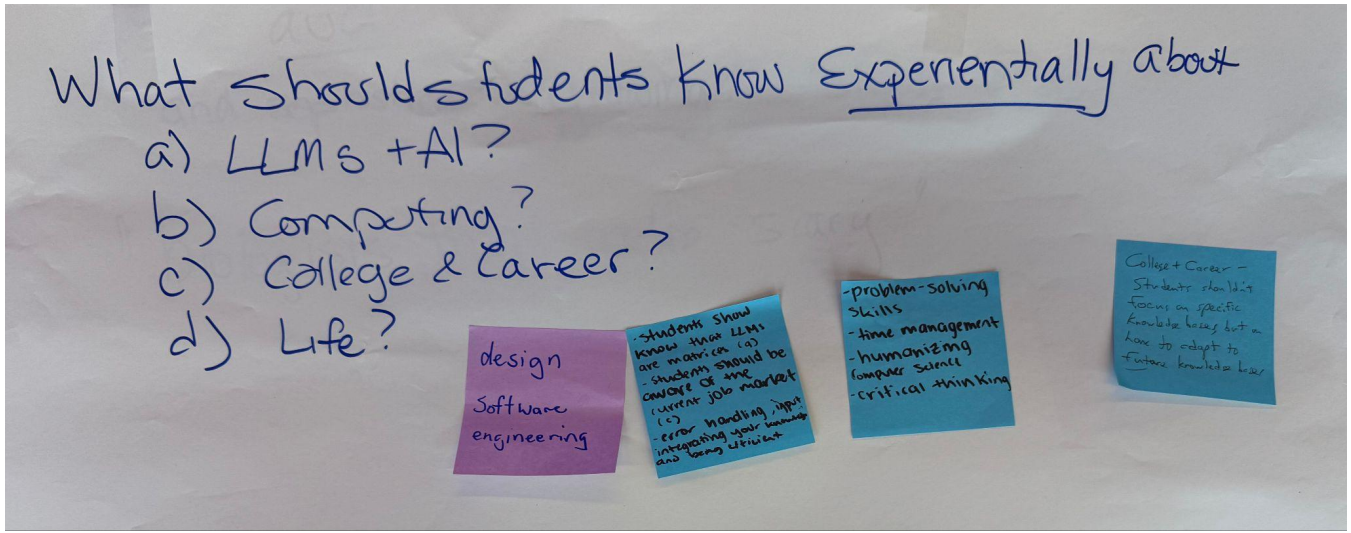

What is being taught now that wont be relevant in the future?
What do you feel you're forced to include?
Shell Scripting? (there's a story...)
current JavaScript framework e.g. REACT


What skills/topics will be more important in the future than now?
Embracing failure/recovering from failure
Interdisciplinary teaching
Communication (oral)


What do we value now that will have lasting value well beyond what's acknowledged by the current hype cycle?
Systems thinking
Growth mindset
Multi/Inter Disciplinary Collaboration
Systems thinking
How to human
Community working thru Imposter syndrome


What else should we be talking about?!?
Classes beyond intro?
"vibe to value"
Ethical implications of AI


Is the audience for computing skills and sophistication growing? Should it be?
"biologists think we're scary"


What should students know Experientially about
a) LLMs + AI?
b) Computing?
c) College & Career?
d) Life?
design
Software engineering
- Students should know that LLMs are matrices (a)
- Students should be aware of the current job market (b)
- error handling, input integrating your knowledge and being efficient
- problem-solving skills
- time management
- humanizing computer science
- critical thinking
College + Career – Students should focus on specific knowledge bases but on how to adapt to future knowledge bases

The CS Core Curriculum Rebalances Knowledge, Skills, and Dispositions

John Stratton, Whitman
strattja@whitman.edu

What does the ACM say now?

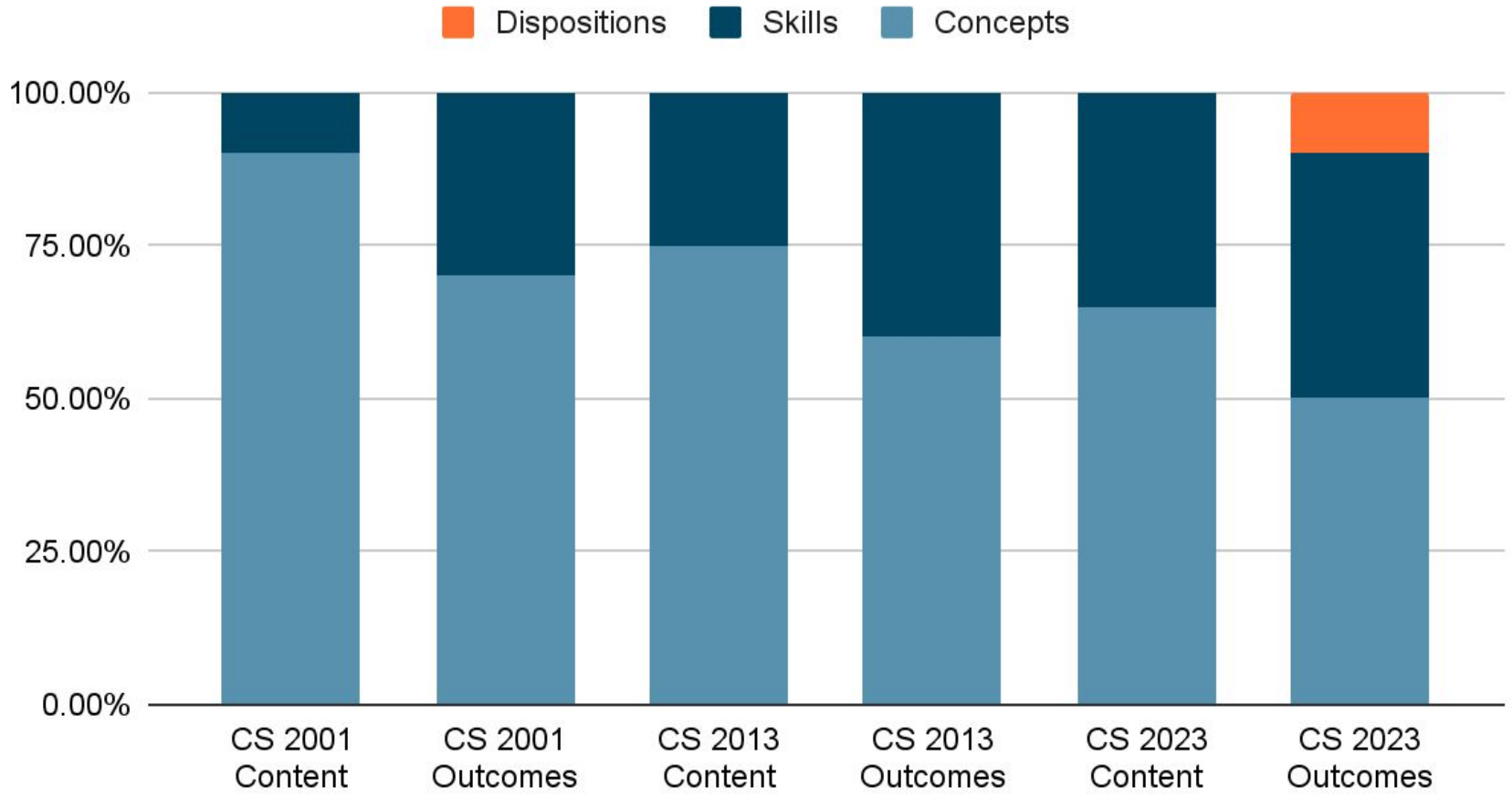
Details of the knowledge areas are in Section 3 of this report. In CS2023, knowledge areas have been expanded to include [ ] in addition to knowledge units:

Knowledge area = { Knowledge units } + { [ ] }

A. Concrete Skills
B. Professional Dispositions
C. Learning Outcomes
D. Professional Skills

Professional dispositions are malleable values, beliefs, and attitudes that enable consistent [, desirable] behaviors

ACM CS Curricula Emphasis (Eyeball test)



Social >> Interactive

Many technologies have been used to make learning more interactive

- Great for teaching concepts
- Good for practicing skills

Dispositions are about values, beliefs, and attitudes, which (I believe) are ***socially constructed****,* and will/should be explicitly integrated into curriculum design.

Curricular space must be ***made*** for this to be added, but it’s worth it.

Whitman's new CS curriculum

- Cut the amount of required ***concepts*** substantially
  - E.g. operating systems implementation, advanced algorithms, automata theory
- Spiraled our introduction of ***skills***
  - e.g. task and project management, collaboration, testing + debugging
- Made room for explicitly addressing ***dispositions***
  - Finding and using references
  - Persistence without spinning, help without dependence

Next steps...

if	then
your program is proactively -- and *radically* -- expanding access	chat with Lauren!
you're looking to emphasize *mindsets as much as concepts*	chat with John!
your department is considering using student-directed learning	chat with Julie!
you'd like to redirect your time to more meaningful tasks (at scale)	chat with Adam!
you're thinking of making computing - or AI - *required*	chat with Zach!

Slides

More slides!

here in the not-scheduled section,
we're all welcome to copy from other favorite slide-creation sources, as well...

Claire!

~~Uncertain~~
Author!