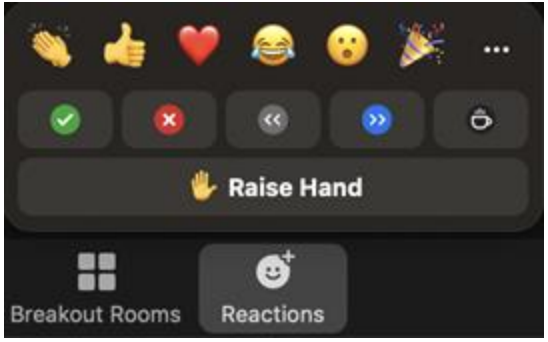
What is going on CS ed?

A crash course on teaching programming

--> Part 1: Classroom techniques.
--> Part 2: Classroom environment

(this is a reaction-friendly space)



Why is learning to program so hard?

$$x = x_0 + vt + \frac{1}{2}at^2$$

```
for each time step:
    make the velocity the sum of current vertical velocity + acceleration
    make the Y position the sum of the current Y position + vertical velocity
    move to the new position
```

B.L.Sherin.A comparison of programming languages and algebraic notation as expressive languages for physics. International Journal of Computers for Mathematical Learning, 6:1–61, 2001. DOI: 10.1023/A:1011434026437

Guzdial, Mark. "Learner-centered design of computing education: Research on computing for everyone." Synthesis Lectures on Human-Centered Informatics 8.6 (2015): 1-165.

“Learning both programming and some subjects (like mathematics or physics) may create some synergy. Learning both together may take less time than learning each separately, because there may be concepts in common (like variables). However, **learning both certainly will take more time than learning only one**.”

Guzdial, Mark. "Learner-centered design of computing education: Research on computing for everyone." Synthesis Lectures on Human-Centered Informatics 8.6 (2015): p.48.

PHYSICS IS COLLABORATIVE

CMS EXPERIMENT, CERN

DEVELOPER POSITIONS — INDUSTRY

- Design
- Frontend
- Backend
- Databases
- UX
- Search

PHYSICS EDUCATION RESEARCH

PROBLEM SOLVING SKILLS AND CONCEPTUAL UNDERSTANDING

- Active learning
- Interactive / group work
- Iterative, standards-based grading
- whiteboarding
- Think-Pair-Share

What is Pair Programming?

- Collaborative learning method in which students program in pairs instead of individually.
- Improves college students' programming competency and increases the likelihood that both male and female students become and remain computer science majors.
- Students work in tandem at one computer while completing regular programming assignments. The "driver "controls the mouse and keyboard while the "navigator" makes suggestions, points out errors, and asks questions.
- The partners routinely switch roles to gain the benefits of each role.

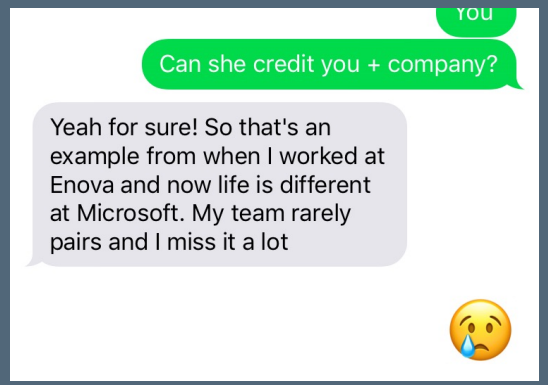
https://www.ncwit.org/resources/pair-programming-box-power-collaborative-learning

Some research results

- **Using collaborative learning research to enhance pair programming pedagogy**
  https://dl.acm.org/citation.cfm?id=1113381
- **Undergraduate student perceptions of pair programming and agile software methodologies: verifying a model of social interaction**
  https://collaboration.csc.ncsu.edu/laurie/Papers/Agile2005.pdf

- Also, pair programming exists in the workplace for many of the same reasons.

Pair programming in the workplace



- Pair programming has been one of the most useful tools to keep our projects going. We usually have the least experienced dev be the driver (person typing). Least experienced can mean the most junior or less familiar with the code base. Both engineers look at the same screen and take turns typing if one person is losing focus or getting tired. While counterintuitive, having two engineers working on the same task helps keep the focus on the problem at hand and avoid slowdowns to check for syntax and other minor validations. Pair coding also encourages relevant knowledge sharing. A few months ago my team was developing a service to aggregate credit reports. I had a big task validating requests from the clients and packaging responses from third-party data vendors. The validations required the use of regular expressions. My coworker and I paired to make sure edge cases for the regular expression were appropriately handled. Countless times one person would remember something the other wouldn't or share an IDE tip that we didn't know about. In one day we not only finished the task but we pretty much developed the backbone of the whole service. After that, we broke up to write unit and end to end tests. The pairing sessions made the entire process much smoother and we finished well ahead of schedule. - Deliana at Enova

Pair programming in the workplace

- The best pairing experience I had was when I first started. We had two keyboards, two mice, one display. We took turns "driving." As you write code, you have to sort of narrate your thoughts so the other person can follow your thought process. That alone is a skill in itself.

  Usually we did the sort of "driver/navigator" interaction, but sometimes we would do a different one called "ping pong" where one person (usually the more senior engineer) writes a failing test that captures the needed functionality, and then the other makes the test green. That goes hand in hand with TDD.

  The last thing that was really helpful was using the pomodoro method while pairing. Pairing is really intense, and doing 25 minutes on, 5 minutes off is helpful to keep you fresh and gives you a chance to use the bathroom or check your phone.

- -Craig at Ultimate Software

Pair programming in the workplace

- We don't officially pair program, but many times on things we want to make sure more than one person knows about early on before we have a chance to document the solution, we'll pair up. The person not 'coding' will often act as a researcher looking up things and proposing ideas while the person coding is manipulating data, testing, writing code.
- -Geo at Ruvos

- Very similar to our active learning methods from physics education research.
- In class, workshop style:
  - Randomly assign partners (not pedagogically backed).
  - Run script to switch who is “driving” at given intervals of time.
  - Green/red post-its indicate who needs help and who is done.
  - Encourage students who are done to help those that need it.

- Often working through Jupyter notebooks with explicit instructions.
- In my classes:
  - Physics classes: pair program in the classroom or lab.
  - Intro classes: NO programming outside of the classroom or solo
  - Upper-level: at-home coding consistent with homework rules.
  - CS classes – pair program for ALL work, in class and homework.
  - Pair programming often preceded by whiteboard activities.

ACTIVITIES

In classroom:

- pair programming

- with rotation

- (love for whiteboard activities too!)

DISCUSS

- **Learning goals:**
  - develop an awareness of the workflow of code development
  - understand the necessity of proper code documentation
  - gain experience with collaborative coding tools
- **Learning outcomes:**
  - write and test a piece of a larger program
  - use, test, and debug code that you did not write

Observations

- Students overall appreciate and enjoy the in-class pair programming.
- Learning in the class is evident.
- Solo programming would be hard to support in the classroom.
- Gives students confidence for solo work or homework.
- Eases difference in coding abilities.

ACTIVITIES

A scientist, Edward Hubble, has brought our team data that he thinks may indicate that our universe is expanding (!!). Our task is create a program that calculates the rate of expansion of the universe from his data. The analysis code must be open source and the results must be clear and reproducible by a third party (reviewer #2 is a notorious skeptic on the expanding universe).

My AI Policies in Classes

Class-created AI Policy: coding

Approach (always in flux, figuring it out):

1. Before class: Read paper/article on LLMs (depends on class): article about Stochastic Parrots or original paper
2. Read syllabus: make sure you understand the language in the learning objectives/outcomes
3. divide into small groups:
   - Briefly discuss learning objectives/outcomes. Is everyone interpreting them the same way? Discuss
   - What AI policies will you employ to ensure you are meeting *my* learning objectives/outcomes for you? Whiteboard
   - How do the policies support/uphold the learning objectives/outcomes? Whiteboard
   - Translate your groups conclusions onto shared Google Doc
   - Read through other groups' ideas. Note similarities/differences. Ask questions.
   - emoji "voting" on policies. Class-wide discussion on concerns [help me with this]. I highlight some of my opinions.
   - boil down to single list
   - everyone agrees to list.
   - Google doc get frozen. Placed on top and prominent on LMS
4. Follow class-defined rules

Class-created AI Policy: coding

Agree: 😃 disagree💀 :Questionable:🤔

Group 1

Let it spot you but you gotta train till failure(convergence)

Yes:

- For errors/ how-to's for coding 😃 😃😃😃😃😃😃
- Helping with plotting results you have gotten 😃🤔😃😃😃😃😃
- Easy stuff/light work 🤔🤔😃😃😃🤔
- Proofreading reports 😃😃 😃😃😃😃😃

No:

- Generating project solutions 😃😃 😃😃😃😃
- Generating reports 😃😃 😃😃😃😃
- Heavy lifting😃 🤔😃😃😃🤔

Group 2

Yes:

- Debug/ask about error messages 😃😃😃😃😃😃 😃😃😃😃
- Format LaTeX 😃😃😃😃 😃😃😃😃😃😃😃
- Explain high level concepts 😃😃😃😃 😃😃😃😃😃
- Help search documentation 😃😃😃😃 😃😃😃😃😃
- Generate short portions of code (must include comment of what is generated)😏😃 🤔🤔🤔🤔🤔🤔🤔
- Produce test cases if needed 😃😃😃 🤔🤔😃😃
- Correct grammar in our paper 😃😃😃😃 😃😃😃😃😃

No:

- Generate large portions of code😃 😃 😃😃 😃😃
- Write the paper😃😃😃 😃😃

Buzzerbeater

Group 3

Yes:

- Search for specific syntax 😃😃😃 😃😃
- Brainstorm idea 😏😃🤔

CSC 374 AI Policy – Projects

Spring 2025

💀💀💀💀💀💀💀💀💀💀💀💀💀💀💀💀💀💀
💀💀💀💀💀💀💀💀💀💀💀💀💀💀💀💀💀💀

We must:

- Report: Note how AI was used in writing as the last section of report
- Code: add comments to generated code blocks saying how generated

Yes:

- Search documentation
- Explain error message
- Explaining general concepts
- Proofreading reports
- Formatting LaTeX
- Generate short portions of code (must explain in short what was generated/fixed)
- Produce test cases if needed

No:

- copy/paste project guidelines so that it will generate the solution
- Generate any part of project reports
- Debugging deep-rooted issues in code and pasting solutions

AI Policy: no coding

Procedure

Each problem: **[10 points]:**

- **[2 points] Initial try:** Using *only* the book and class notes as a resource, attempt to solve the problem. This is graded on an authentic attempt, not correctness. Show *all* work.
- **[2 points] Learning:** Open the provided solutions. Evaluate your work using the template provided below. Keep in mind that official solutions often skip steps and are not representative of all the work you should have shown. Solution manuals often contain mistakes. This is graded for accurate analysis of work.
  - *Markup:* Simple X/check/emoji of your **initial try**.
  - *Evaluation:* Short (1-3 phrase) summary of your initial understanding
  - *Edit/Rework:* Based on your *Markup*, indicate whether you will edit or rework the problem.
- **[4 points] Solution:** This is graded on the correctness and completeness (all necessary steps shown) of your solution. See examples provided for structure of your solution based on your **Initial try**. You can use *any* resources to complete the solution, but it MUST reflect your understanding of the problem. Resources include: Dr. Kuchera, classmates, LLMs, the provided solutions, YouTube videos, etc. If provided a very similar question, you should be able to apply what you learned here. This is graded for correctness and thoroughness of showing work.
- **[2 points] Understanding:** These are points you assign yourself based on your understanding of the problem. ***Note that I have the right to adjust these points based on your* **Solution**.
  - *[0 points]*: At the end of this problem, I only vaguely understand the concepts here. I am not confident that I can successfully answer a related question on a closed book quiz.
  - *[1 point]:* At the end of this problem, I understand the concepts and some details, but do not think I can fully reproduce this work in a closed book quiz.
  - *[2 points]:* At the end of this problem, I am confident that I will successfully answer a related question on a closed book quiz.