Pair Programming

Luther Tychonievich

A single practice that

- Reduces teacher work
- Increases student enjoyment
- Increases student retention
- Increases student learning
- Appeals to girls & boys
- Enables niftier projects
- Increases employability

Structured Cooperation

- Two (2) people, one (1) focus
- Defined roles: Driver + Navigator
  - 1 driver, rest navigators
  - driver: the immediate (code, syntax)
  - navigator: the big picture (design, logic)
- Role switching protocols:
  give take timer milestone

Activity

- 1 paper, 1 pen
- Draw an outdoor scene
- Include at least 3 of
  - pet, livestock, farmer, children, river, trees, hills, crops, house, sunset, fantasy element

Forming Pairs

- I know of studies that suggest

◦ Random	◦ Orchestrated
◦ Similar ability	◦ Diverse ability
◦ Homogeneity	◦ Heterogeneity
◦ Self-selection	◦ Personality Type

- No clear winner

  (middle school: pair within gender)

Soloists and Experts

- Pro solo:
  - Uncooperative is bad for everyone
- Pro pair anyway:
  - Cooperation important skill
  - Ego ≠ ability
- Suggestion: decide case-by-case

Tips for Students

- `http://youtu.be/rG_U12uqRhE` (or others)
- Talk & Listen
- Patience & Respect
- Breaks
- Cleanliness, Breath, ~~Perfume~~, Personal Space
- Don’t be intimidated

Tips for Teachers

- Control switching
- Unsolicited feedback on dynamic (navigation)
- Large project, open upper bound
- Take Qs from pair, not person
- Pairs “break;” can work through, re-pair, make triples, or make solo

Grading (1 of 2)

- Post-survey: “% of work you/partner did”
  - Scale grades accordingly
  - Disagree? Talk with each

  - Some skip this step…
  - …some increase its complexity

Grading (2 of 2)

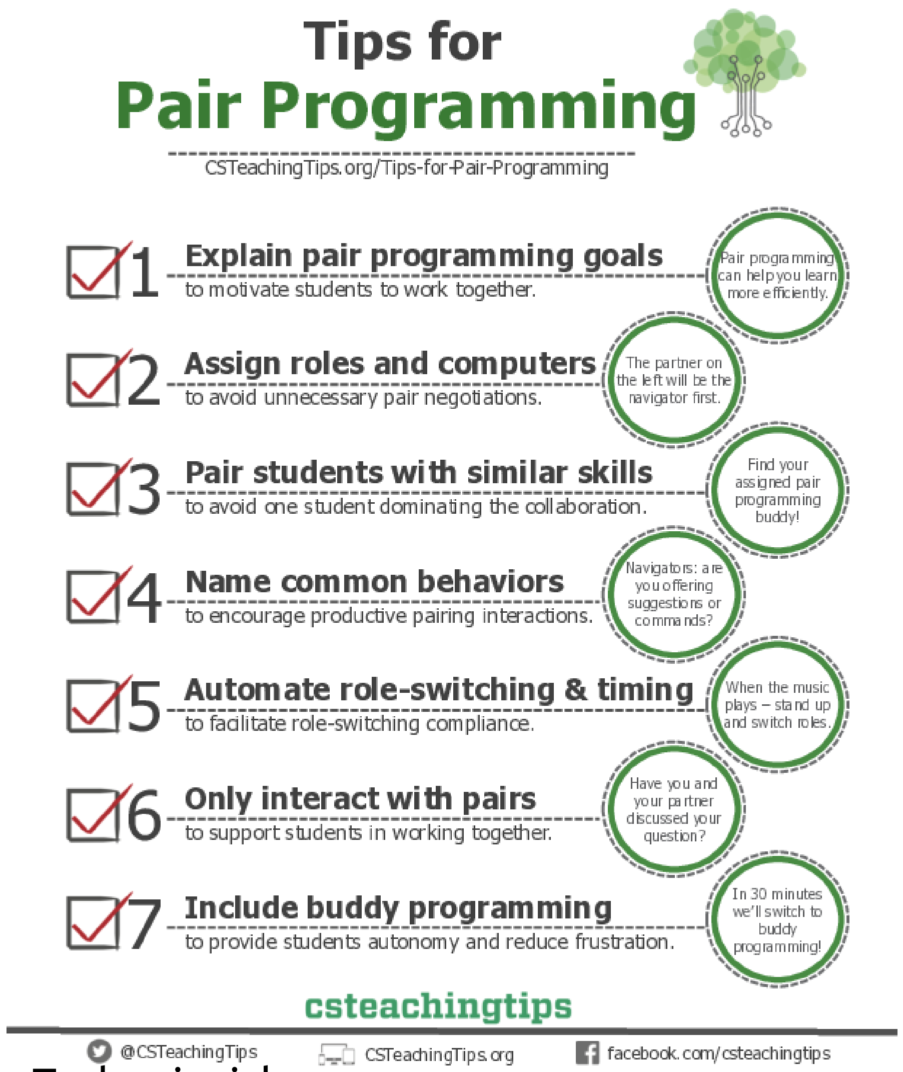
- Mix solo, pair, and group work
  - Individual assessment
  - Think-pair-share
  - Simultaneous feedback (clickers)
- Only pair large-enough projects
  - Each should drive 2+ times

Caveats

- Tolerate increased volume
- Projects need to be ~2× larger
- More human management
- Change details at your own risk

Caveats

- Research consistent: it works
- But experience suggests:
  - More bias incidents
  - Devolves into divide-and-conquer
- *Must* be actively supervised!



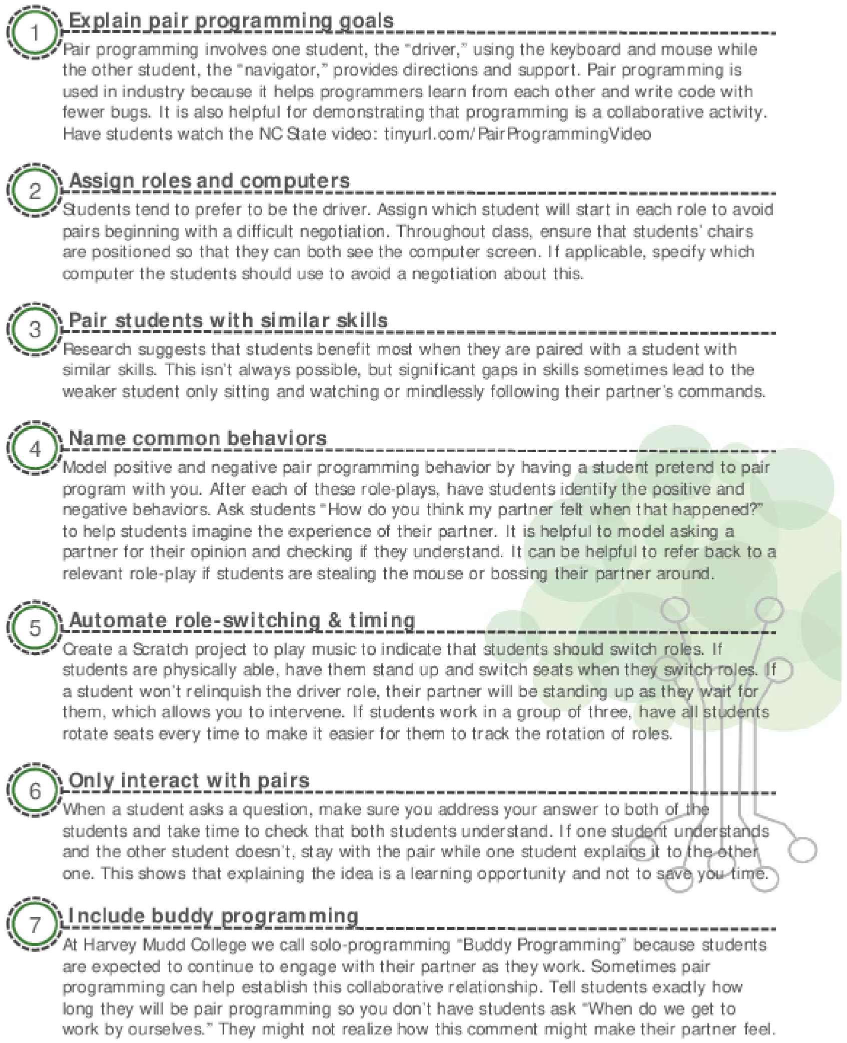
Tips for Pair Programming

CSTeachingTips.org/Tips-for-Pair-Programming

1. **Explain pair programming goals** to motivate students to work together.
   - Pair programming can help you learn more efficiently.
2. **Assign roles and computers** to avoid unnecessary pair negotiations.
   - The partner on the left will be the navigator first.
3. **Pair students with similar skills** to avoid one student dominating the collaboration.
   - Find your assigned pair programming buddy!
4. **Name common behaviors** to encourage productive pairing interactions.
   - Navigators: are you offering suggestions or commands?
5. **Automate role-switching & timing** to facilitate role-switching compliance.
   - When the music plays – stand up and switch roles.
6. **Only interact with pairs** to support students in working together.
   - Have you and your partner discussed your question?
7. **Include buddy programming** to provide students autonomy and reduce frustration.
   - In 30 minutes we'll switch to buddy programming!

csteachingtips

@CSTeachingTips CSTeachingTips.org facebook.com/csteachingtips



1 Explain pair programming goals

Pair programming involves one student, the "driver," using the keyboard and mouse while the other student, the "navigator," provides directions and support. Pair programming is used in industry because it helps programmers learn from each other and write code with fewer bugs. It is also helpful for demonstrating that programming is a collaborative activity. Have students watch the NC State video: tinyurl.com/PairProgrammingVideo

2 Assign roles and computers

Students tend to prefer to be the driver. Assign which student will start in each role to avoid pairs beginning with a difficult negotiation. Throughout class, ensure that students' chairs are positioned so that they can both see the computer screen. If applicable, specify which computer the students should use to avoid a negotiation about this.

3 Pair students with similar skills

Research suggests that students benefit most when they are paired with a student with similar skills. This isn't always possible, but significant gaps in skills sometimes lead to the weaker student only sitting and watching or mindlessly following their partner's commands.

4 Name common behaviors

Model positive and negative pair programming behavior by having a student pretend to pair program with you. After each of these role-plays, have students identify the positive and negative behaviors. Ask students "How do you think my partner felt when that happened?" to help students imagine the experience of their partner. It is helpful to model asking a partner for their opinion and checking if they understand. It can be helpful to refer back to a relevant role-play if students are stealing the mouse or bossing their partner around.

5 Automate role-switching & timing

Create a Scratch project to play music to indicate that students should switch roles. If students are physically able, have them stand up and switch seats when they switch roles. If a student won't relinquish the driver role, their partner will be standing up as they wait for them, which allows you to intervene. If students work in a group of three, have all students rotate seats every time to make it easier for them to track the rotation of roles.

6 Only interact with pairs

When a student asks a question, make sure you address your answer to both of the students and take time to check that both students understand. If one student understands and the other student doesn't, stay with the pair while one student explains it to the other one. This shows that explaining the idea is a learning opportunity and not to save you time.

7 Include buddy programming

At Harvey Mudd College we call solo-programming "Buddy Programming" because students are expected to continue to engage with their partner as they work. Sometimes pair programming can help establish this collaborative relationship. Tell students exactly how long they will be pair programming so you don't have students ask "When do we get to work by ourselves." They might not realize how this comment might make their partner feel.

Industry

- Part of Xtreme Programming, other agile
- Used throughout development
- Pair vs. two independent:
    - Pair ~15% less productive
    - Pair ~85% higher quality
- Coding Dojos, Mob Programming
    - 1 driver, *n* navigators

Resources

`cstapestry.wikidot.com/pair-programming`

- NCWIT Pair Prog. in a Box
- Online group management tools
- Video for students to watch
- Websites with current research
- Other slides we’ve used for this in the past

A single practice that

- Reduces/changes teacher work
- Increases enjoyment, retention, learning
- Appeals to girls & boys
- Enables niftier projects
- Increases employability
- But gets mixed feedback in classroom practice