

Pair Programming in CS1: Plenty to gain and nothing to lose *

Charlie McDowell
Computer Science Department
University of California, Santa Cruz

Linda Werner, UCSC CS Dept.

Brian Hanks, Univ. of Colorado, Fort Lewis, CS Dept.

Heather Bullock and Julian Fernald
UCSC Psychology Dept.

David Draper and Milovan Krnjajic
UCSC Applied Math and Statistics Dept.

* This work was supported by the National Science Foundation.



What is Pair Programming?

- Two programmers working at a single workstation.
- It is not divide and conquer.
- It is part of the eXtreme Programming software methodology.



Why Use It In Teaching?

- PP is a form of collaborative learning, and research shows that collaborative learning is good.
- PP allows an instructor more time per program when grading.
- PP makes fewer demands on TAs because students answer each other's questions.
- **PP results in higher confidence and higher retention**



Why Not Use Pair Programming?

- Students need to learn by doing it themselves. (If learning to program is like learning to ride a bike, you can't learn by watching someone else do it.)
- How will an instructor know who really did the work?
- Dealing with intra-pair student conflict is a lot of work.

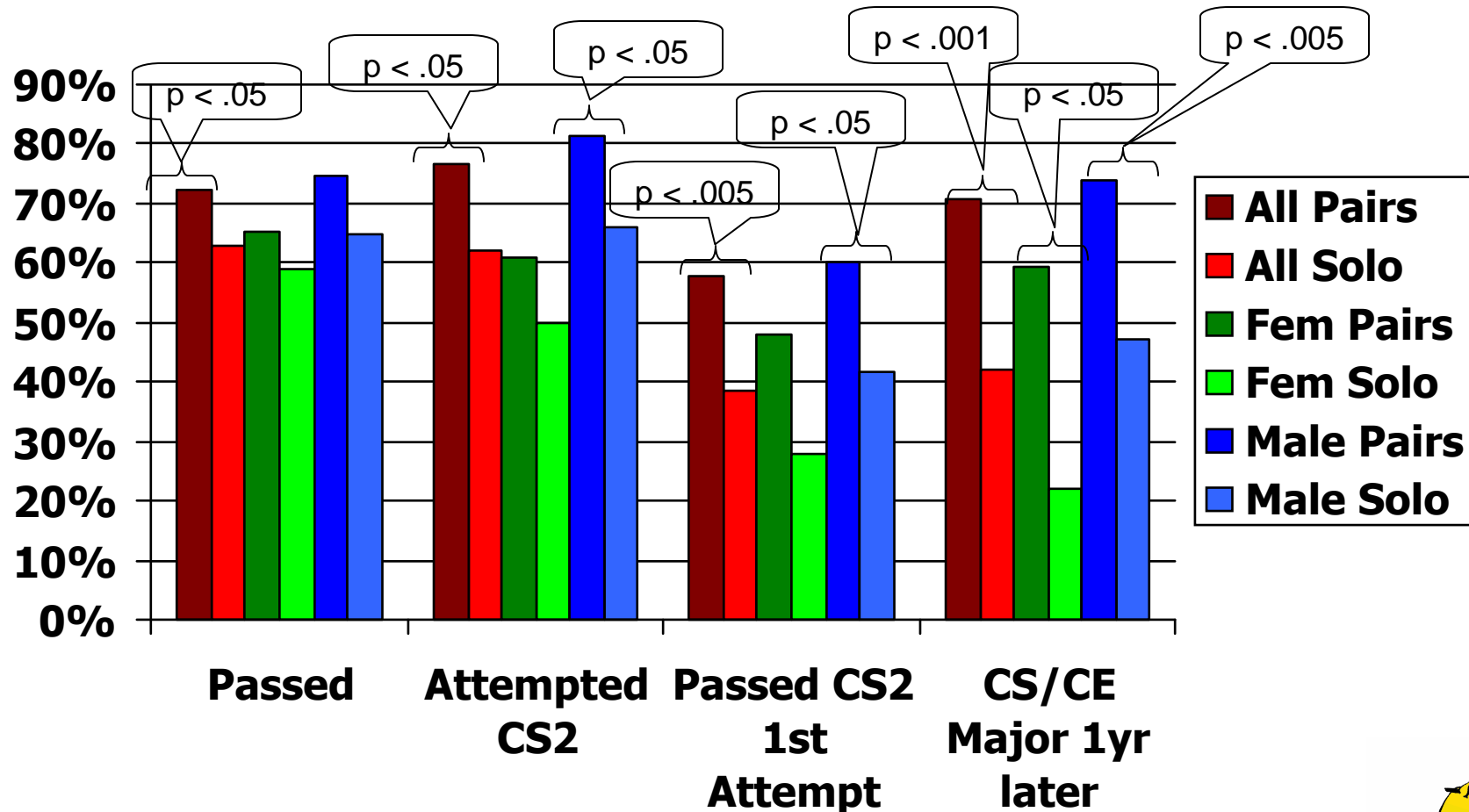


Three Studies

- 2000-01 (3 pairing sections 1 non-pairing)
 - 554 students (141 women, 413 men)
 - UCSC
- 2003-04 (4 pairing sections 4 non-pairing)
 - 214 students (41 women, 173 men)
 - SJSU 124 students
 - Cabrillo College 90 students
- 2003-04 (3 pairing sections)
 - 115 students (24 women, 91 men)
 - UCSC



Pair Programming in CS1 UCSC 00-01



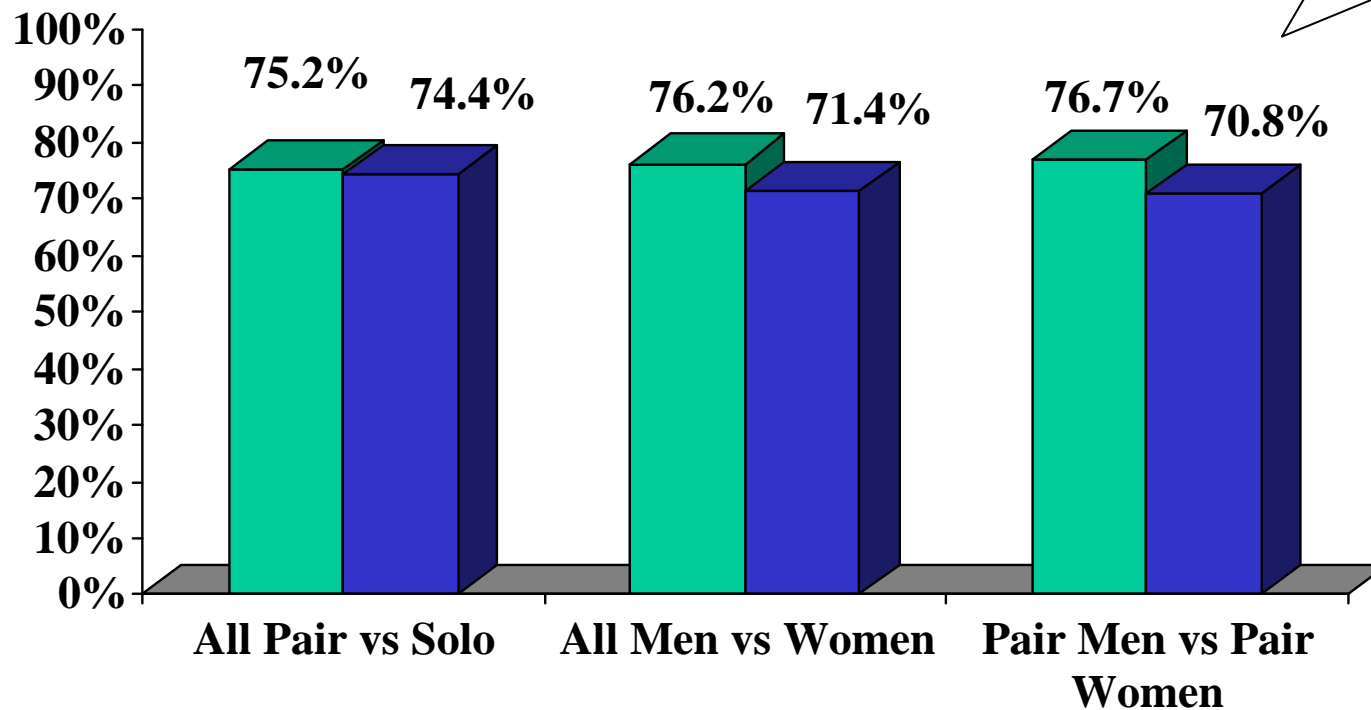
Statistically significant differences are indicated with relevant χ^2 p values.



No Difference in Final Exam Scores

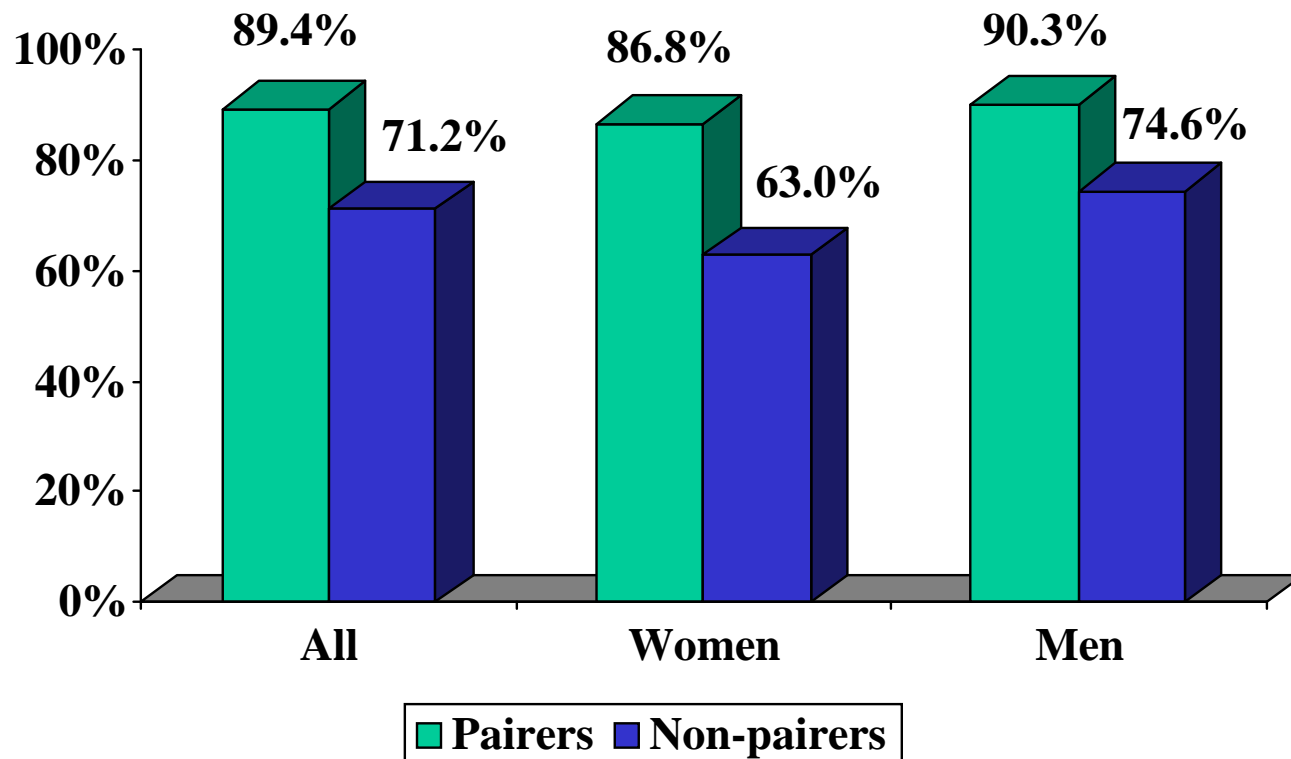
Approaching significance, $p = .09$, goes away when SAT Math used as covariate

Significant, $p < .01$, goes away with SAT Math covariate.



Confidence in Solution

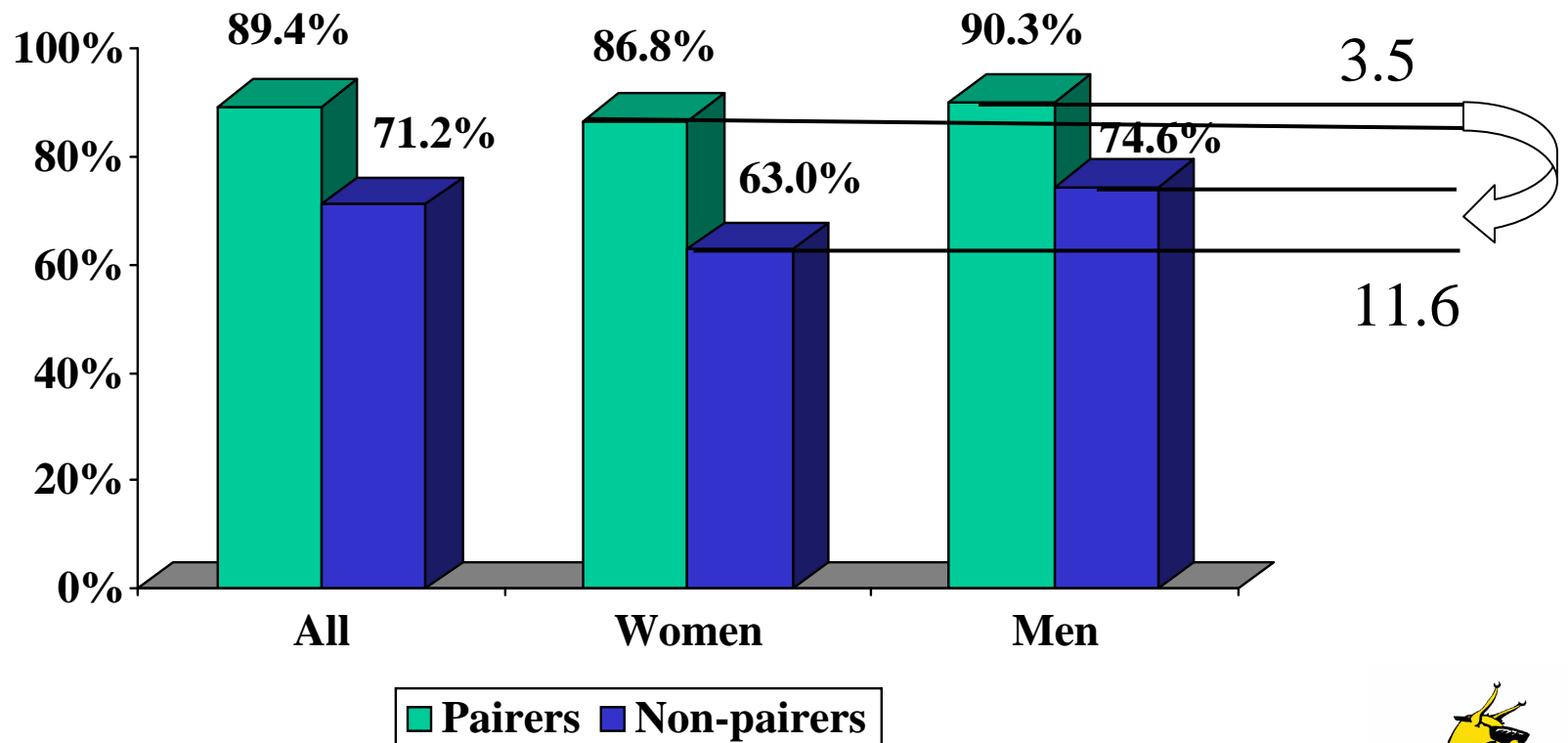
All differences were significant, $p < .001$



Closing One Gender Gap

Confidence in Solution

Gender gap with pairs is 3.5%. For non-pairs it is 11.6%.



Primary Study Conclusion

- Students working in pairs produce better programs.
- Students working in pairs are less likely to drop.
- Students working in pairs do at least as well on the exams.
- Students working in pairs are more likely to attempt CS2 and equally likely to pass it the first time.
- Students working in pairs are more likely to be a declared CS/CE/ISM major 1 year later.
- The relative increase in confidence and retention was greater for women than for men.



Two Follow-up Studies

- 2000-01 (3 pairing sections 1 non-pairing)
 - 554 students (141 women, 413 men)
 - UCSC
- **2003-04 (4 pairing sections 4 non-pairing)**
 - **214 students (41 women, 173 men)**
 - SJSU 124 students
 - Cabrillo College 90 students
- **2003-04 (3 pairing sections)**
 - **115 students (24 women, 91 men)**
 - UCSC



Follow-up Conclusions

- Smaller numbers generally resulted in no statistically significant results.
- Trends generally supported prior findings.
 - more pairers took the final
 - more pairers passed the class
 - exam scores were the same
 - pairers were more confident
 - pairers produced better programs
- One SJSU professor refused to use non-pairing after 1st semester.



Pair Programming Enforcement

Which of the following best describes your success with pair programming on this assignment?

- a) My partner and I successfully followed the **pair programming guidelines**. In particular each partner drove roughly 50% of the time we were working together, and at most 25% of my individual effort for the assignment was spent working alone. Any work done by a solitary programmer was reviewed by the other partner.
- b) I tried to work with a partner but had problems as explained in the next question.
- c) I worked alone.



Who is learning more?

(Who is having more fun and will continue?)



Talk Ended here

- The following slides are some extras that I had if I had extra time or for use in response to questions.



Results: Questions Addressed

- Did they complete the class (take the final)?
- Did they pass with a C or better?
- What was their average programming project score?
- What was their final exam score?
- How confident were they in their programming project solutions?
- How satisfied were they with the programming process?
- How much did they enjoy the programming process?
- Did they subsequently take CS2 and if so, did they pass it?
- What was their major one year later?



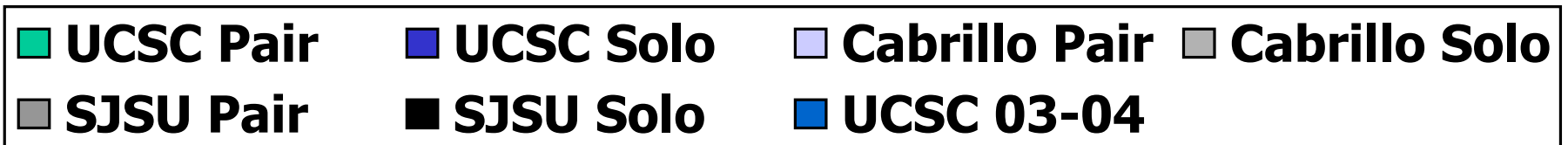
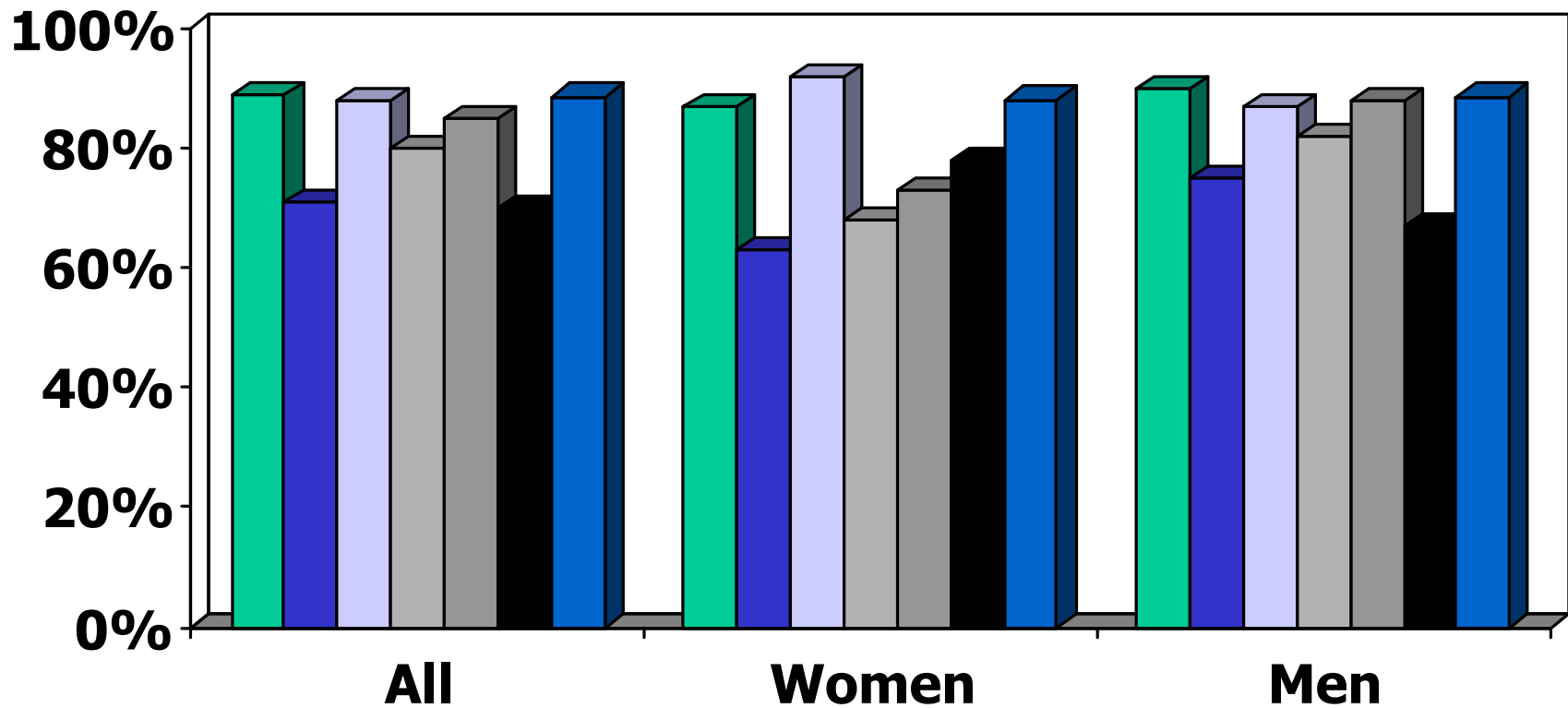
Confidence, Satisfaction, Enjoyment

- On a scale from 0 (not at all confident) to 100 (very confident), how confident are you in your solution to this assignment?
- (Pairs) How satisfied are you with the way that you and your partner worked together on this assignment? (1 = very dissatisfied, 7 = very satisfied)
- (Solo) How satisfied are you with how you spent your time on this assignment? (1 = very dissatisfied, 7 = very satisfied)
- How much did you enjoy working on this programming assignment? (1 = not at all, 7 = very much)



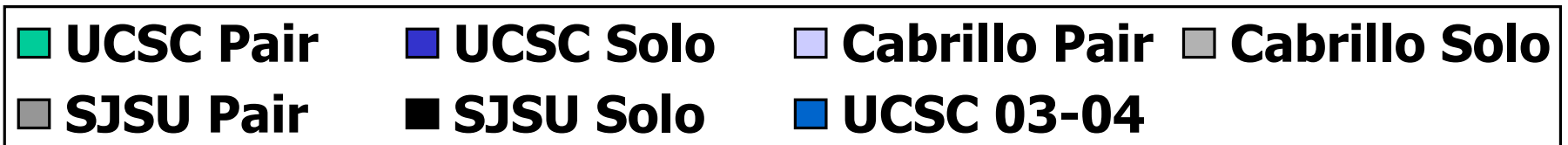
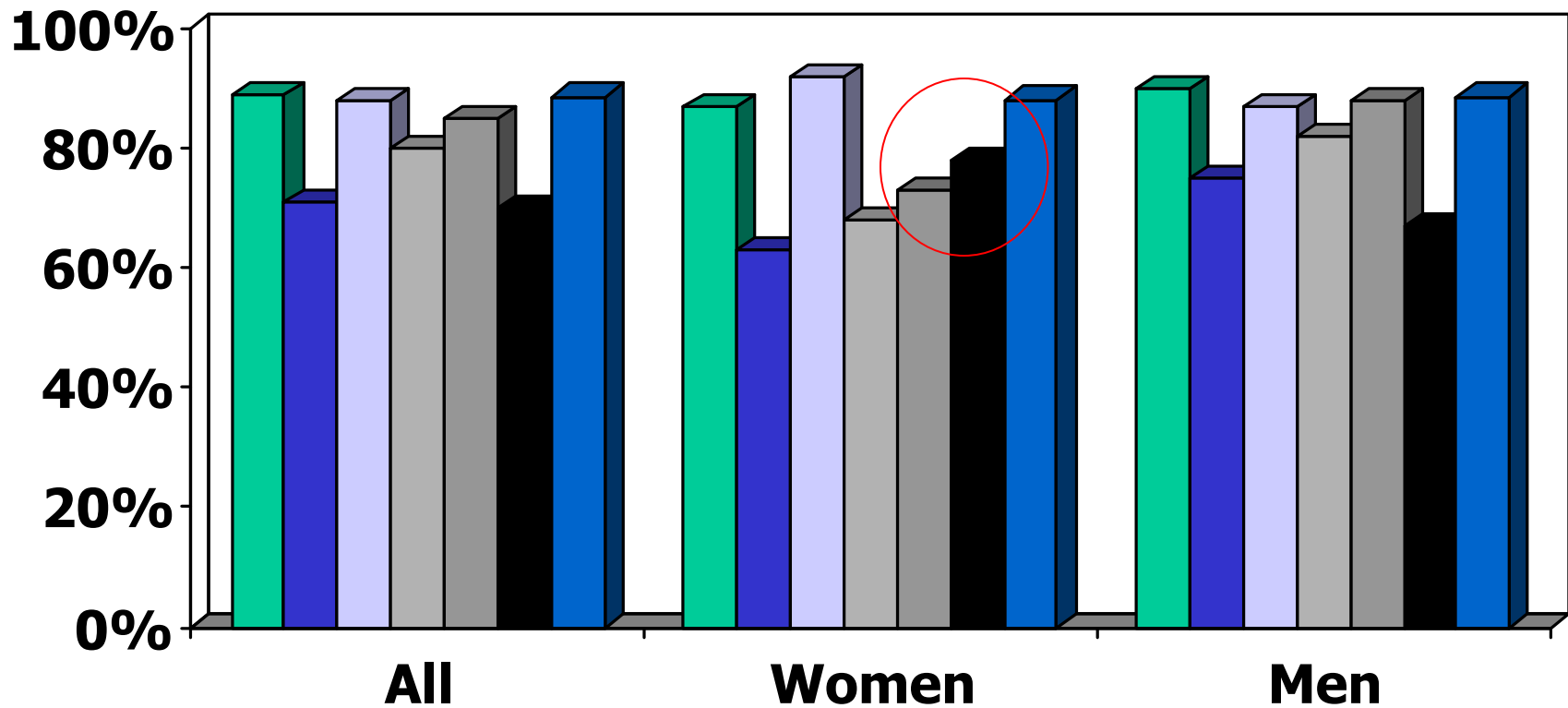
Confidence in Solution

Pairers report greater confidence



Confidence in Solution

Pairers report greater confidence



Many Questions Remain

- Should students change partners during a semester/quarter? If so how often?
- What about a partner that is uncooperative?
- Does this work better for some students than others? If so, which ones? Can we make it work for all?
- Did students in fact learn more? Do the exam scores reflect the true value (or weaknesses) of pair programming?



What About the Freeloaders in Pairs?

- CS2 is programming intensive and did not allow pairing.
- Pairing students from CS1 passed CS2 at comparable rates to non-pairing students.
- This would suggest that the pairing students overall learned to program (as required by CS2) just as well as the non-pairing students.



No Difference in Exam Scores?

- There is no significant difference.
- Does that mean pair programming had no affect on overall student performance as measured by the exams?
- We believe that pair programming may have increased student performance, but this was obscured by the variation in the number of students that dropped prior to the final exam.

