Enjoyment: How much did you enjoy working on this programming assignment? (1=not at all, 7=very much)

Paired students enjoyed working on programming assignments (M=5.15) more than non-pairing students (M=4.69), Likewise, among just the males, paired students reported greater enjoyment (M=5.23) than non-pairing students (M=4.75). Women paired students also reported greater enjoyment (M=4.90) than non-pairing women (M=4.65), however this difference was not significant. There was also no significant difference in the reported enjoyment of all women versus all men.

In addition to enjoying their coursework more, students who paired reported significantly higher confidence in their program solutions (89.4%) than students who worked independently (71.2%). Consistent with findings in other areas, men were significantly more confident (87.0%) than women (81.1%). There was also a significant interaction between pairing and gender with regard to reported confidence. Follow-up tests of the interaction indicated that pairing resulted in more confidence for both women (86.8% vs. 63.0%) and men (90.3% vs. 74.6%). However, the 24% increase in confidence that pairing afforded women was even greater than the 15% confidence boost experienced by men who had the benefit of pairing. The result was a significant decrease of a gender gap in confidence as shown in the figure here.

Conclusion

The results of this study provide some of the most compelling evidence to date of the effectiveness of pair programming as a pedagogical tool. It appears that pairing bolsters course completion and consequently course pass rates, and contributes to greater persistence in computer science-related majors. Moreover, students who paired were more likely to pass the subsequent programming course that required them to work alone. This is a strong indicator that pairing did not result in a significant number of students passing the course without learning how to program due to a "free ride" from their partner. The pairing students also produce higher quality programs, are more confident in their work, and enjoy it more. We hope these findings will encourage instructors to use pair programming not only in their introductory courses, but also in their upper-level courses.

The continued underrepresentation of women in computer science underscores the need for strategies that foster women's interest and promote their success [5]. Pair programming appears to be one such approach [6]. That the benefits associated with pair programming extend to both men and women speaks to its broad-based appeal. As we continue to investigate the effects of this technique on attracting and retaining female students, parallel research investigating these phenomena in the workplace is also needed. C

This work was funded by National Science Foundation grant EIA-0089989. Any opinions, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Beck, K. Extreme *Programming Explained: Embrace Change.* Addison-Wesley, Reading, MA, 2000.
2. Hanks, B. and McDowell, C. Program quality with pair programming in CS1. In *Proceedings of the 9th Annual Conference on Innovation and Technology in Computer Science Education.* (Leeds, UK, 2004), SIGCSE Bulletin, 176–180.
3. McDowell, C., Werner, L., Bullock, H., and Fernald, J. The effects of pair-programming on performance in an introductory programming course. In *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education* (KY, 2002) 38–42.
4. McDowell, C., Werner, L Bullock, H., and Fernald, J. The impact of pair programming on student performance and pursuit of computer science related majors. In *Proceedings of the 25th International Conference on Software Engineering. IEEE Computer Society* (Portland, OR, 2003), 602–607.
5. *Tech-Savvy Educating Girls in the New Computer Age.* 2000. American Association of University Women Education Foundation; Executive summary at www.aauw.org/research/techexecsumm.cfm.
6. Werner, L.L., Hanks, B., and McDowell, C. Pair-programming helps female computer science students. *J. Educational Resources in Computing 4,* 1 (2005).
7. Williams, L.A. and Kessler, R.R. All I really need to know about pair programming I learned in kindergarten. *Commun. ACM 43,* 5 (May 2000), 108–114.
8. Williams, L.A. and Kessler, R.R. The effects of "pair-pressure" and "pair-learning" on software engineering education. In *Proceedings of the 13th Conference on Software Engineering Education and Training. IEEE Computer Society* (Austin, TX, 2000), 59–65.
9. Williams, L., Kessler, R., Cunningham, W., and Jeffries, R. Strengthening the case for pair programming. *IEEE Software 17,* 4 (2000), 19–25.
10. Williams, L., McDowell, C., Nagappan, N., Fernald, J., and Werner, L.L. Building pair programming knowledge through a family of experiments. In *Proceedings of the IEEE International Symposium on Empirical Software Engineering.* (Rome, Italy, 2003), 143–153.
11. Women, Minorities and Persons with Disabilities in Science and Engineering. NSF. 2004; www.nsf.gov/statistics/women.

Charlie McDowell (charlie@cs.ucsc.edu) is a professor in the Computer Science Department at the University of California, Santa Cruz.
Linda Werner (linda@cs.ucsc.edu) is a lecturer in the Computer Science Department at the University of California, Santa Cruz.
Heather E. Bullock (hbullock@cats.ucsc.edu) is an associate professor in the Psychology Department at the University of California, Santa Cruz.
Julian Fernald (jfernald@cats.ucsc.edu) is the director of Institutional Research in the Psychology Department at the University of California, Santa Cruz.