

were significantly more likely to have declared a computer science related major than non-pairers (56.9% vs. 33.8%), and that was the case for both men (59.5% vs. 41.1%), and women (46.3% vs. 11.1%). See the table here for results.

**Course performance.** In addition to completion and pass rates, we looked at specific parts of the course performance. We measured two related, but distinct indicators of course mastery. The first, the quality of student programs, was operationalized as students' normalized average score on the graded programming assignments. For the second indicator, the extent to which students could apply the concepts covered during the course, we used final exam scores, which all students took independently regardless of whether they paired or not. For the following analyses we included only the 486 students who completed the course. Completing the course is defined as taking the final exam.

Among students who completed the class, those who paired produced significantly better programs (86.6%) than those who worked alone (68.1%). There was no significant gender difference in average programming scores (men's and women's scores were 81.9% and 82.5% respectively), nor was there an interaction between gender and pairing. In other words, pairing was associated with significantly higher scores for both women and men.

It may be the reluctance of some computer science faculty to use pair programming in their classes is due to a concern that at least some students will "earn" grades that predominantly reflect their partner's work. It is possible, for example, that the pairing students in our study earned higher average programming scores simply because weaker students received scores that were primarily due to the work of the stronger student in the pair, thus artificially inflating the average programming scores of the pairers.

Elsewhere we have argued that the very process of working collaboratively enhances the quality of programs that pairs produce [3]. In that paper we compared two sections of an introductory programming course taught by the same instructor, and for which assignments were intentionally designed to be equivalent. We found the average score on programming

assignments of students in the pairing section was significantly higher than the average score of the top 50% (based on final exam scores) of the non-pairing section.

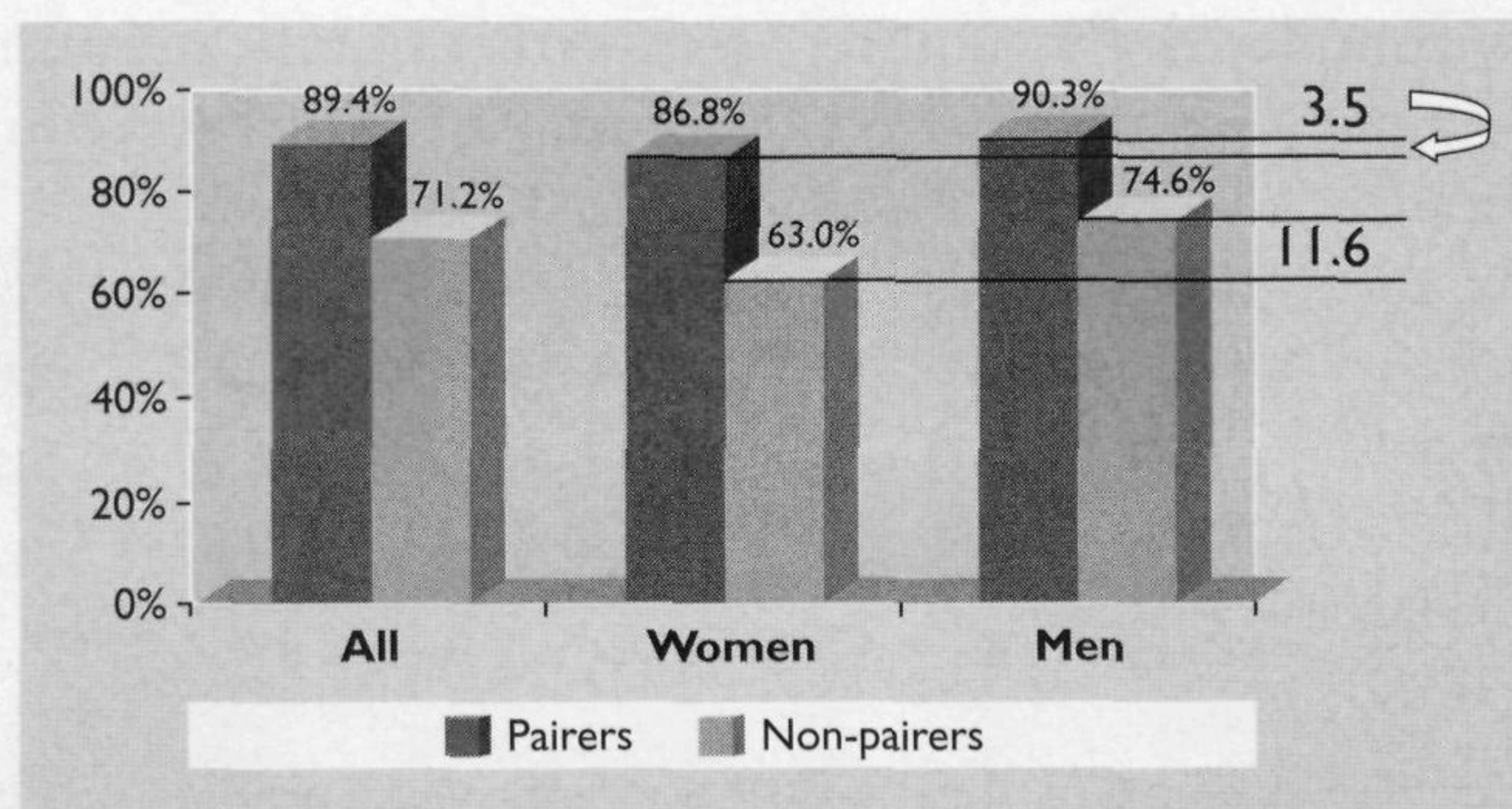
Because students in different sections of this study did not complete exactly the same programming assignments, we did a follow-up study [2] in which a section of pairing students was given the same programming assignments as the non-pairing students from our original study. In that study we again found that the programs produced by the pairing students were significantly better than those produced by the

non-pairing students, although the difference was not as great as in the original study. This suggests that some of the difference reported in our main study could be from variations in the difficulty of the assignment, but that the overall conclusion is unchanged.

Because all of the students in this study took the final exam independently, we considered final exam scores to be a strong indication of the extent to which students had mastered the course material. There was no significant difference in the average final exam of the pairers (75.2%) and the non-pairers (74.4%). This finding strongly suggests a student's ability to independently apply concepts to novel problems is not compromised by learning to program in pairs. Indeed, considering that a significantly greater percentage of the students who paired took the final, it seems that learning to program in pairs results in mastery for a greater percentage of students.

**Confidence and enjoyment.** Of course the most important goal for students in any class is mastery of the material. This is certainly the case for introductory programming courses, where future success is dependent on a strong foundational knowledge. However, subjective experiences in introductory programming courses may also contribute to decisions about whether to pursue computer science-related degrees. For this reason it is important to understand how the experience of pairing influences students' confidence and enjoyment of their work. Students responded to the following questions in their logs completed after each graded programming assignment.

*Confidence:* On a scale from 0 (not at all confident) to 100 (very confident), how confident are you in your solution to this assignment?



Confidence in program solutions—closing one gender gap.