

# Programming Practices: Pair Programming

Norman Jacobson, Sr. Lecturer (Ret.)  
Dept. of Computer Science, UC Irvine  
Tapestry Workshop 2014  
July 31, 2014  
[jacobson@uci.edu](mailto:jacobson@uci.edu)

# Briefest History

An aspect of agile programming (1990's)

Adapted for education in early 2000's

Suggested for use in ICS in 2005

(I took some convincing!)

First integrated into Fall 2006 intro course

Used in many CS courses nationwide —  
because it works!

# In essence, pair programming is

two (student) programmers:

a “driver” who performs the “on computer” tasks

&

a “navigator” who watches and comments,

working together at one computer,

switching roles every several minutes.

# Why Do It?

## Students Do Better!

Pair programming...

- does not disadvantage students, even the few who “hate” working with another
- improves the quality of students’ programs and learning, compared to students working alone — two heads really are better than one
- enables solidifying and deepening of students’ understanding through interaction with a peer, and through teaching another
- provides a peer who can immediately address questions and difficulties
- provides students a direct, immediate opportunity to read another programmer’s code, a skill not manifest when programming individually
- increases the number of students who successfully complete the course
- when used in CS1, leads to increased solo programming competency in CS2

# Why Do It?

## Increased recruitment and retention in CS majors

Pair programming...

- for most students, is more enjoyable than programming alone
- tends to engage students more actively and directly in the course
- increases students' — especially female students' — confidence with programming
- increases the number of introductory students who stay in computing majors
- increases the percentage of introductory students, especially women, who declare a computer science major

# Why Do It?

## Additional Benefits

### Pair programming...

- gives students "an excuse" to get to know each other, helping to form peer support groups
- provides students experience in working in a collaborative manner
- encourages students to turn to their partners as the first option to obtain help with their programming; this rapid assistance improves productivity, increases a sense of accomplishment, and reduces reliance on course staff
- typically improves students' work ethic — they don't want to let their partner down
- provides practice in fairly reviewing and evaluating others
- is increasingly being used in higher education and in industry

# Common Objections to Adoption...

## ...and Overcoming Them

- We don't have a closed computer lab
  - reserve a corner of an open one
  - use (students' own) laptops/tablets in the classroom
- It's difficult to access an individual's programming competency
  - not if the graded instruments are programming exams that are individually taken
- Students won't switch roles unless forced; not switching negates pair programming's benefits, while enforcement takes inordinate staff time
  - yes, staff needs to prod students to switch, but it takes virtually no time at all
- Ensuring workable pairings requires too much instructor time to be feasible
  - voluntary pairings, with staff guidance, almost always work out, and are quick to establish
  - same for pairing students of similar work ethic
  - (other approaches seem not to work very, or the jury is still out)
  - staff can, in reasonable time, pair up the few students who have difficulty finding partners
- "Broken" partnerships occur so often that pair programming in effect would not occur
  - very few partnerships (< 5% in our experience) have compatibility problems
  - requiring students to evaluate each other encourages them to "play nice"
  - appropriate staff intervention typically enables the partnership to continue productively, if not perfectly
- Solo programming in subsequent courses is significantly impaired
  - it may seem so, but the data strongly say otherwise

# Fine Summaries

## *Pair Programming*

from the *Encyclopedia of Software Engineering*

See especially sections 5 (Pair Programming in an Educational Setting)  
and 7 (Principles of Pair Programming)

[http://collaboration.csc.ncsu.edu/laurie/Papers/ESE  
%20WilliamsPairProgramming\\_V2.pdf](http://collaboration.csc.ncsu.edu/laurie/Papers/ESE%20WilliamsPairProgramming_V2.pdf)

## *Lessons Learned from Seven Years of Pair Programming at North Carolina State University*

[ftp://ftp.ncsu.edu/pub/unity/lockers/ftp/  
csc\\_anon/tech/2007/TR-2007-20.pdf](ftp://ftp.ncsu.edu/pub/unity/lockers/ftp/csc_anon/tech/2007/TR-2007-20.pdf)

**Laurie Williams**

North Carolina State University  
Department of Computer Science



# National Center for Women and Information Technology's Pair Programming in a Box

Provides a turn-key set of materials for an instructor  
to quickly and easily implement pair programming  
in a programming course

*[http://www.ncwit.org/resources/pair-programming-box-  
power-collaborative-learning](http://www.ncwit.org/resources/pair-programming-box-power-collaborative-learning)*

# Pair Programming Works Even Better If You

- Enable and encourage *Peer Instruction*
  - don't grade the assignments — students are then free to get help from any source, including their peers, without fear of cheating
  - use peer tutors
- Keep assignments small enough to complete in lab, in no more than a few weeks
  - it's often difficult for students to meet out of class, so don't require that they do
  - assignments can build toward more complex and interesting programs
- Have assignments liberally use images/animation/sound (if not too disruptive) — *Media Computation*
  - much more engaging than text-based work
  - much easier to debug
  - instructor-provided libraries/classes/routines can hide complexity
- Switch partners every assignment or two
  - helps students to learn to play well with others
  - minimizes the stress of a difficult partnership

# PP in a Box in More Detail

## Pair Programming At-a-Glance

### Overview of

- Basics

- Benefits

- Activities

- Pair formation

- Adjusting the Course

## Integrating Pair Programming into the Course

- Lab sections

- Preparing students and course staff

- Assessing programming competency

- Pairing students

- Dealing with a faltering pair

- Forming new partnerships

- Questionnaires and surveys

- Awarding points for pair programming activities

# PP in a Box in More Detail

## Sample Handouts for an Introductory Course

- Pair Programming in CS1 (for students)
- Pair Programming Guidelines for Course Staff
- Pair Programming Partner Questionnaire
- Sample Student Survey

## Resources – 43 references to papers, videos, tutorials, reading lists

- Overviews
  - Pair programming in introductory computing courses
  - Pair programming in the classroom
- Compatible pairs
- Lab exams
- Preparing staff
- Evaluating Pairs
- Resource lists on the web
- Materials for students