Studying the Effects of Adding 'In-Lab' Programming Tests to a CS Service Course

Edwin M. Knorr and Christopher Thompson
Department of Computer Science, UBC
knorr@cs.ubc.ca

Background

- CPSC 259 is a core course for Electrical Engineering students who are *not* in the Computer Engineering option.
  - Computer Science students cannot take it.
  - It is taught by the Computer Science department.
- It is a course in:
  - Intermediate C Programming
  - Data Structures
  - Basic Algorithms
- Enrolment:
  - 197 students in Jan-Apr 2013
  - 191 students in Jan-Apr 2014

Programming Assessments and 2014 Intervention

- **2013**
  - **No in-lab programming tests**
  - Lots of programming homework (done in pairs)
  - Programming questions on the Midterm and Final Exam (done alone, on paper)

- **2014**
  - **5 in-lab programming tests @ 2% each** (done alone, on a lab computer; average = 68.2%, SD = 23.6%)
    - 6 different lab sections; so, we created 30 tests in all
    - Students work under exam-like conditions
  - Lots of programming homework (done in pairs)
  - Programming questions on the Midterm and Final Exam (done alone, on paper)

February 2016: We re-marked 2013’s and 2014’s programming questions (approx. 200 final exams for each year) to provide consistency across years.

Why Use In-Lab Programming Tests in Addition to Paper-Based Exams?

- Students use the same compiler and integrated development environment (IDE) in which they do their programming labs
  - “Real-world” conditions
  - Compiler gives instant feedback
    - Errors, warnings, output from code
  - Students can repeatedly test and debug their code
    - They may not do this extent of “testing” when writing code on paper.
- Increased and spaced testing is known to increase performance in various disciplines (Roediger, *et al.*, 2010)
  - i.e., 5 in-lab tests + 1 midterm + 1 final exam = 7 tests of programming (plus their programming homework)
- We hope it will build student confidence in writing code.
- We hope it will improve the pair-programming contributions for their homework (e.g., keep partners accountable).

Final Exam Results: 2013 vs. 2014

- To determine whether students' programming abilities improved on the final exam, we compared **all common programming-related questions** (identical or isomorphic).
- Surprisingly, there was no statistically significant difference in performance on these questions.

Year	*n*	Mean	Std. Dev.	*p*-value (2013 vs. 2014 means)	Median
2013	190	63.2%	20.3%		65.0%
2014	187	65.7%	22.1%	0.24	67.1%

- Correlation of these final-exam programming-related questions with the combined in-lab programming test scores in 2014:
  - r = 0.71 (n/a for 2013)

Final Exam Results: 2013 vs. 2014 (cont.)

- When comparing just the 10-mark **Linked List programming question:**
- No statistically significant change between 2013 and 2014. (We double-checked the scores; remarkably consistent!)

Year	*n*	Mean	Std. Dev.	*p*-value (2013 vs. 2014 means)	Median
2013	190	61.3%	32.6%		70.0%
2014	187	61.3%	33.0%	0.99	70.0%

- Correlation of this final-exam programming question with the combined in-lab programming test scores in 2014:
  - r = 0.66 (n/a for 2013)

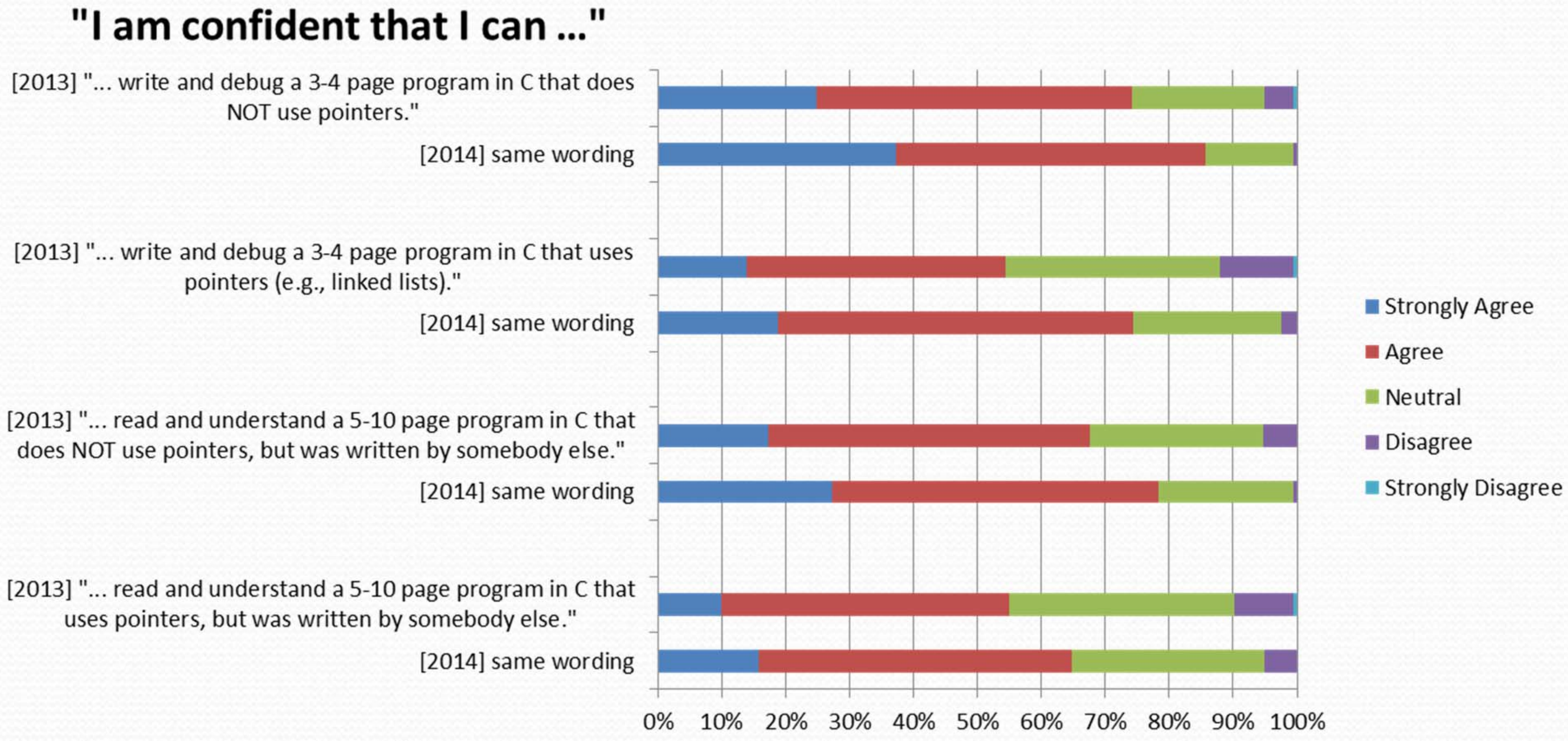
Final Exam Results: 2013 vs. 2014 (cont.)

- When comparing only the 8-mark **Binary Search Tree programming question:**
- No statistically significant change between 2013 and 2014.

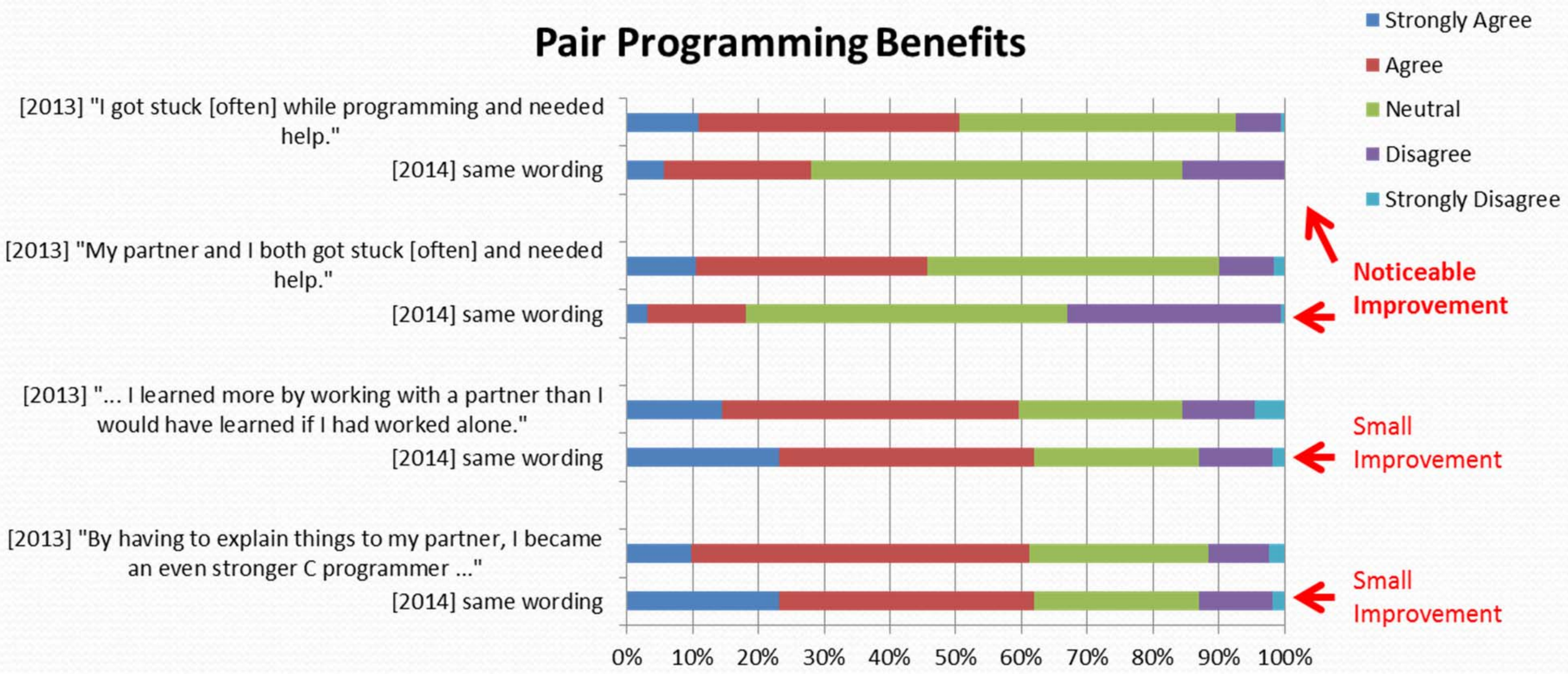
Year	*n*	Mean	Std. Dev.	*p*-value (2013 vs. 2014 means)	Median
2013	190	40.1%	30.2%		37.5%
2014	187	43.7%	33.3%	0.28	37.5%

- Correlation of this final-exam programming question with the combined in-lab programming test scores in 2014:
  - r = 0.57 (n/a for 2013)

Students' Confidence in Programming Abilities (2013 vs. 2014 … End of Course)



Students’ Opinions about Pair-Programming Abilities and Partnerships (2013 vs. 2014)



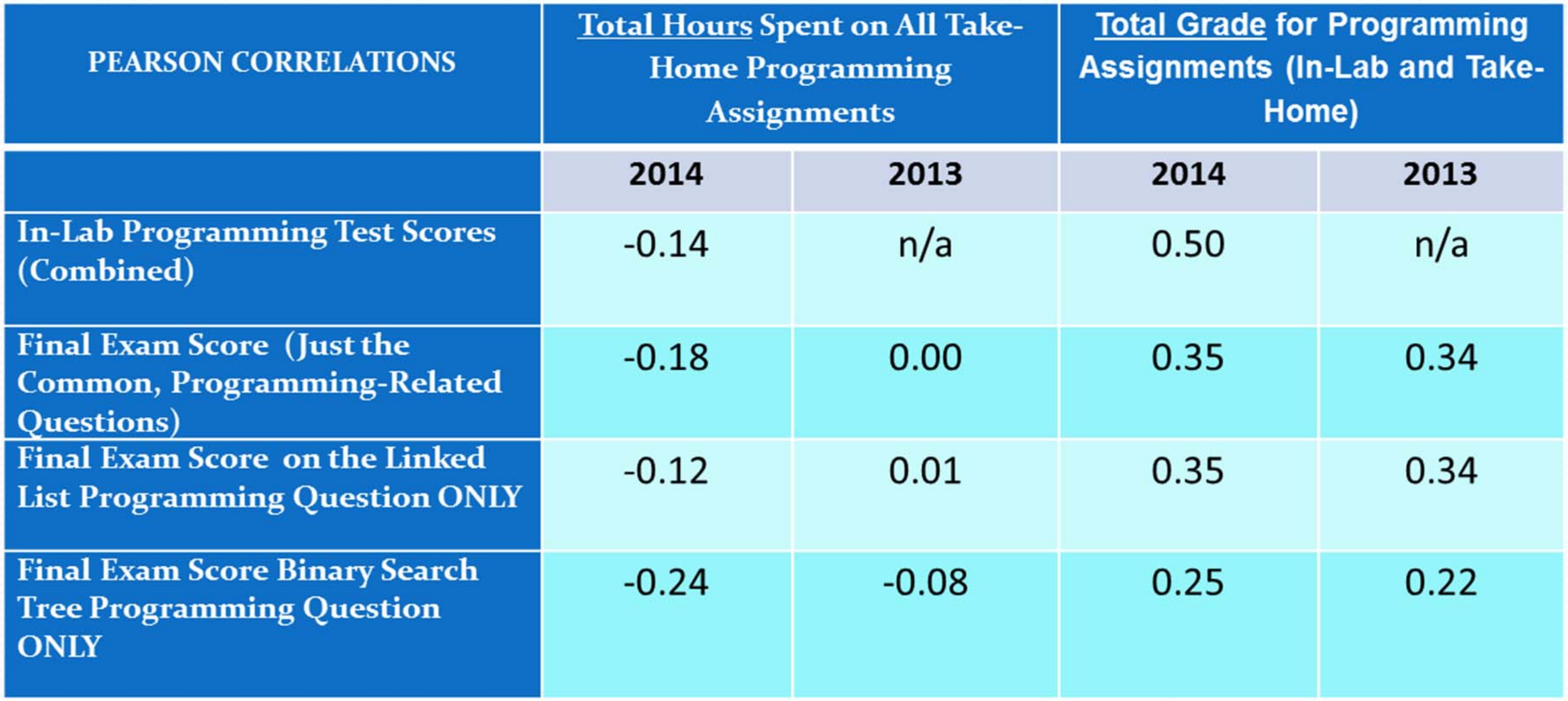
Noticeable Improvement

Small Improvement

Small Improvement

Correlations Involving the Take-Home Programming Homework and Tests/Exams (2014 vs. 2013)

- **Hours Spent**: slightly negative correlation with all tests
- **Grade Achieved**: mild to medium positive correlation with tests



PEARSON CORRELATIONS	Total Hours Spent on All Take-Home Programming Assignments		Total Grade for Programming Assignments (In-Lab and Take-Home)	
	2014	2013	2014	2013
In-Lab Programming Test Scores (Combined)	-0.14	n/a	0.50	n/a
Final Exam Score (Just the Common, Programming-Related Questions)	-0.18	0.00	0.35	0.34
Final Exam Score on the Linked List Programming Question ONLY	-0.12	0.01	0.35	0.34
Final Exam Score Binary Search Tree Programming Question ONLY	-0.24	-0.08	0.25	0.22

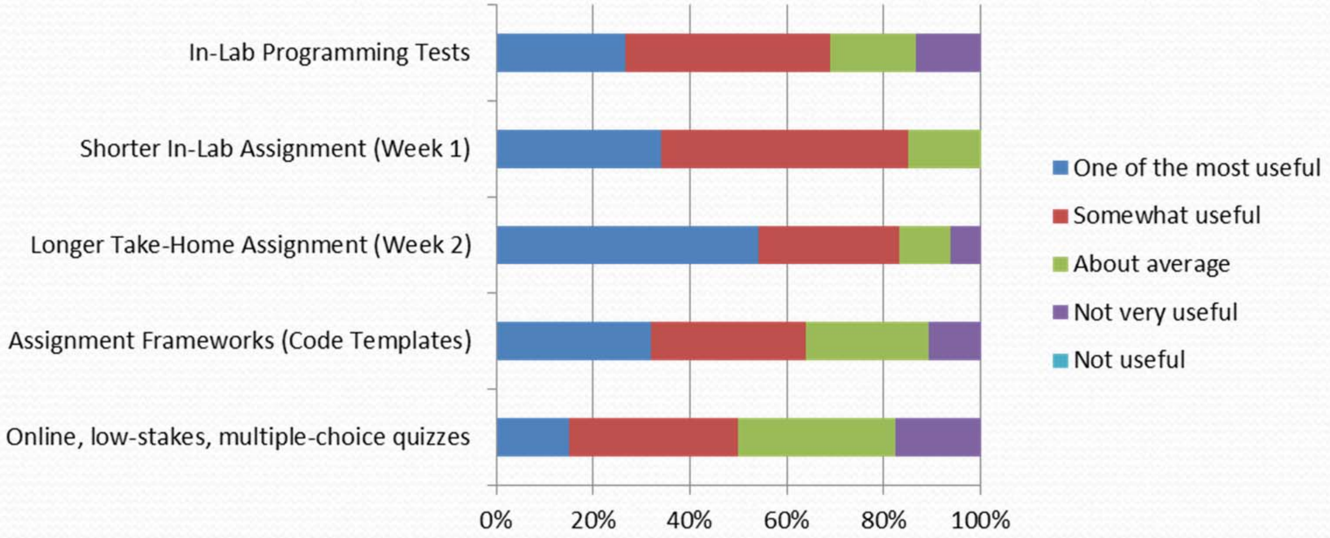
2-Year Longitudinal Survey

- Survey given in Feb 2016 to students from Jan-Apr 2014 (i.e., reflection two years later)
  - 48 out of 188 students responded (26%)

- "Preparing for, and doing, the bi-weekly in-lab programming tests helped me to become a better partner for the take-home programming assignments."
  - 81% agreed

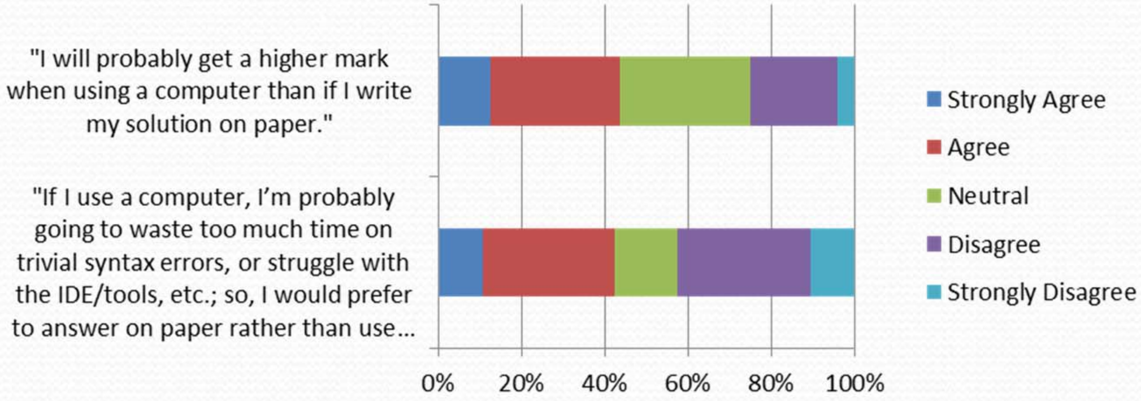
2-Year Longitudinal Survey (cont.)

- "Which of the following learning techniques were most useful in improving your C programming ability?"



2-Year Longitudinal Survey (cont.)

- "On the final exam, suppose you are allowed to use a computer when writing a C function. The computer has an IDE and compiler (e.g., Visual Studio), but no external help is allowed, there's no access to Web pages, and you can't look at any other code. Given an identical amount of time to answer the question on paper or on a computer, how strongly do you agree with these statements?"



2-Year Longitudinal Survey (cont.)

- “Engineering is a zero-sum game: If I put more time into any one of my courses, then I have to take time away from another course.”
    - 74% agreed

Note: Engineering students are already taking a large load (about 3-5 more credits (1.5 courses) per term than a typical Science student), and they don’t want more things to study

Summary and Conclusions

- When going from 2013 (no in-lab programming tests) to 2014 (in-lab programming tests):
  - Students still have trouble performing well on programming questions on the final exam.
    - No statistically significant change from 2013 to 2014
  - Student surveys reveal that confidence has improved.
    - But increased confidence did not result in better performance.
  - Pair programming satisfaction and readiness seems to have improved.

Future Work / Open Questions

- Why are students not improving in programming on the final exam, after doing the in-lab programming tests?
  - Final exam questions too hard?
  - Wrong "kinds" of questions?
  - Differences in final exam environment vs. in-lab environment?
  - Student utility functions?
    - Trade-off of time vs. better grade for a course that's not focused on their area of specialization
    - Issue of sufficient understanding vs. mastery
  - Other variables?
- Which students *were helped* by the in-lab tests?
  - Was there something we didn't assess, that *did* improve?
- Should improved confidence imply better performance?