Effective Closed Labs in CPSC 121: Lessons from Eight Terms of Action Research

Steve Wolfman,
Elizabeth Patitsas,
CPSC 121 Staff

CPSC 121, Brief Overview

Unusual introductory course combining discrete mathematics (theoretical tools) and digital logic (hardware). Required for CPSC majors and a few other programs.

~100-200 students per term. ~6-15 TAs per term. TA background in digital logic often relatively weak. Many different faculty and TAs work on the course over time.

Methodology: Action Research

Our approach is based on action research.

- Goal: to enact social change. We used action research to identify problems in the labs, and to assess changes intended to improve them.
- Process: through empowering the stakeholders through participatory research.
- Role of the researcher: the researcher is a participant in the course; here we act simultaneously as course designers, assessors, and educators.

Methods: Regular + End-of-Term Student Feedback

CPSC 121 Lab Feedback

We are in the process of redeveloping many of the labs in this course to improve them. For one mark in lab each lab, please provide feedback on how you found the lab. Feedback is anonymous - after hitting submit, show the ensuing page to your TA for the mark.

* Required

What is the number of the lab you are reviewin

I had enough time to work on this lab.

1 2 3 4 5

Strongly disagree ○ ○ ○ ○ ○ Strongly agree

The written instructions were clear and well-wri

1 2 3 4 5

Strongly disagree ○ ○ ○ ○ ○ Strongly agree

The lab was relevant to the lecture material.

1 2 3 4 5

Strongly disagree ○ ○ ○ ○ ○ Strongly agree

The lab was interesting.

CPSC 121 End of Term Lab Survey

As part of our endeavour to improve the CPSC 121 labs, we would like your feedback on how the labs were overall this term. Filling the survey is worth one bonus mark in lab 10. Your answers, as usual, are confidential.

The labs contributed to my understanding of the course material.

1 2 3 4 5

Strongly disagree ○ ○ ○ ○ ○ Strongly agree

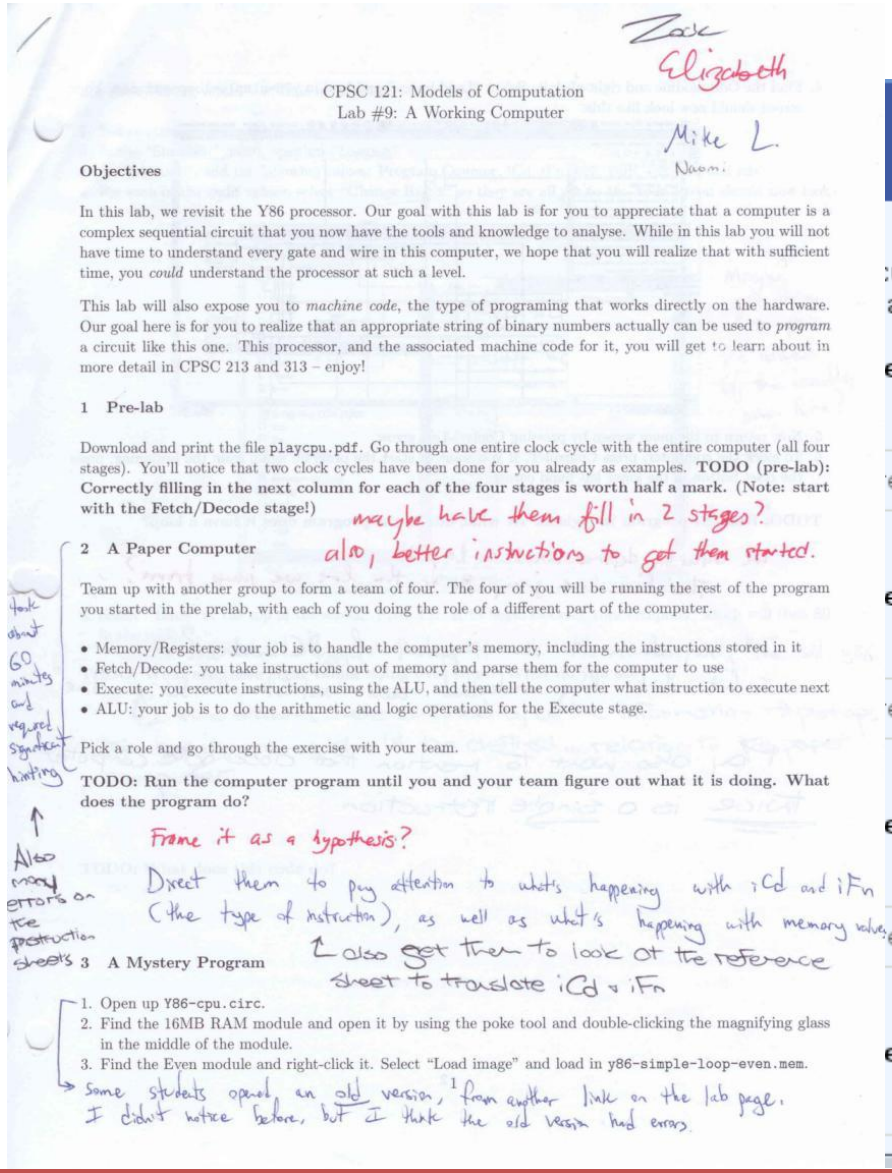
The pre-labs contributed to my understanding of the course material.

1 2 3 4 5

Strongly disagree ○ ○ ○ ○ ○ Strongly agree

The challenge problems contributed to my understanding of the course material.

Methods: End-of-Term TA Reviews



Zack
Elizabeth
Mike L.
Naom.

CPSC 121: Models of Computation
Lab #9: A Working Computer

Objectives

In this lab, we revisit the Y86 processor. Our goal with this lab is for you to appreciate that a computer is a complex sequential circuit that you now have the tools and knowledge to analyse. While in this lab you will not have time to understand every gate and wire in this computer, we hope that you will realize that with sufficient time, you *could* understand the processor at such a level.

This lab will also expose you to *machine code*, the type of programing that works directly on the hardware. Our goal here is for you to realize that an appropriate string of binary numbers actually can be used to *program* a circuit like this one. This processor, and the associated machine code for it, you will get to learn about in more detail in CPSC 213 and 313 – enjoy!

1 Pre-lab

Download and print the file `playcpu.pdf`. Go through one entire clock cycle with the entire computer (all four stages). You'll notice that two clock cycles have been done for you already as examples. **TODO (pre-lab): Correctly filling in the next column for each of the four stages is worth half a mark. (Note: start with the Fetch/Decode stage!)**

maybe have them fill in 2 stages?
also, better instructions to get them started.

2 A Paper Computer

Team up with another group to form a team of four. The four of you will be running the rest of the program you started in the prelab, with each of you doing the role of a different part of the computer.

- Memory/Registers: your job is to handle the computer's memory, including the instructions stored in it
- Fetch/Decode: you take instructions out of memory and parse them for the computer to use
- Execute: you execute instructions, using the ALU, and then tell the computer what instruction to execute next
- ALU: your job is to do the arithmetic and logic operations for the Execute stage.

Pick a role and go through the exercise with your team.

TODO: Run the computer program until you and your team figure out what it is doing. What does the program do?

Frame it as a hypothesis?

Direct them to pay attention to what's happening with iCd and iFn (the type of instruction), as well as what's happening with memory values
↳ also get them to look at the reference sheet to translate iCd & iFn

3 A Mystery Program

1. Open up `Y86-cpu.circ`.
2. Find the 16MB RAM module and open it by using the poke tool and double-clicking the magnifying glass in the middle of the module.
3. Find the Even module and right-click it. Select "Load image" and load in `y86-simple-loop-even.mem`.

Some students opened an old version,[1] from another link on the lab page. I didn't notice before, but I think the old version had errors.

took about 60 minutes and required significant hinting

Also many errors on the instruction sheets

CPSC 121 Lab TA feedback

curriculum development initiative, we'd like to know how you (the TAs) abs.

e interesting for me

	1	2	3	4	5	
ee	○	○	○	○	○	Strongly agree

e fun for me

	1	2	3	4	5	
ee	○	○	○	○	○	Strongly agree

e rewarding for me

	1	2	3	4	5	
ee	○	○	○	○	○	Strongly agree

e of an appropriate difficulty

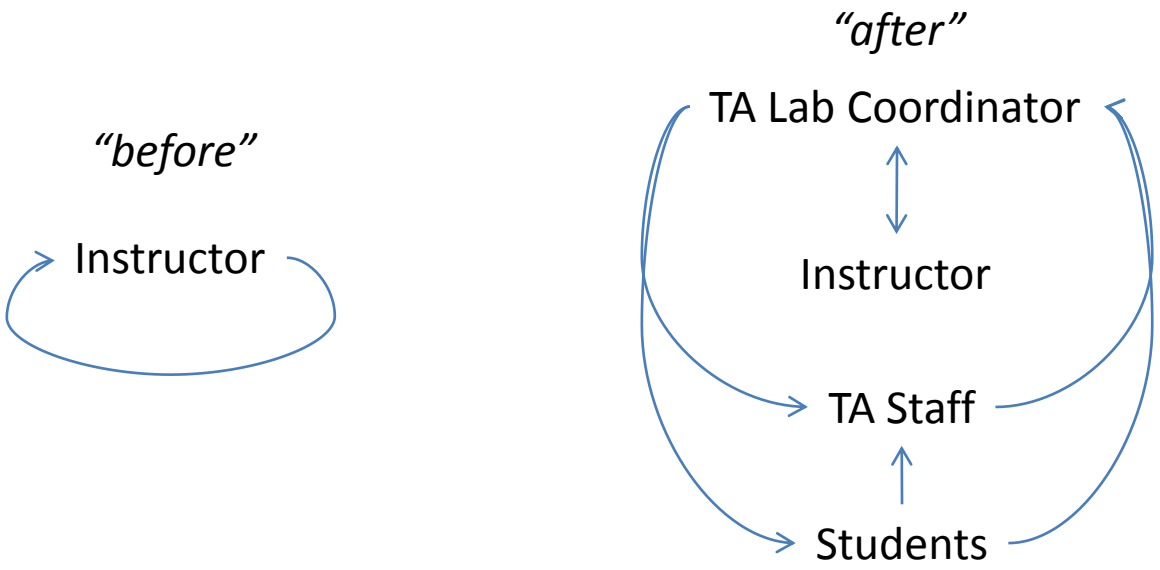
1	2	3	4	5

Additional Methods

- Continuous staff feedback (lab prep/staff meetings)
- Early focus groups (kicked off this effort, finding that students found labs unrewarding/disconnected from lecture)
- Final exam questions
- Midterm and End-of-Term Evalutions

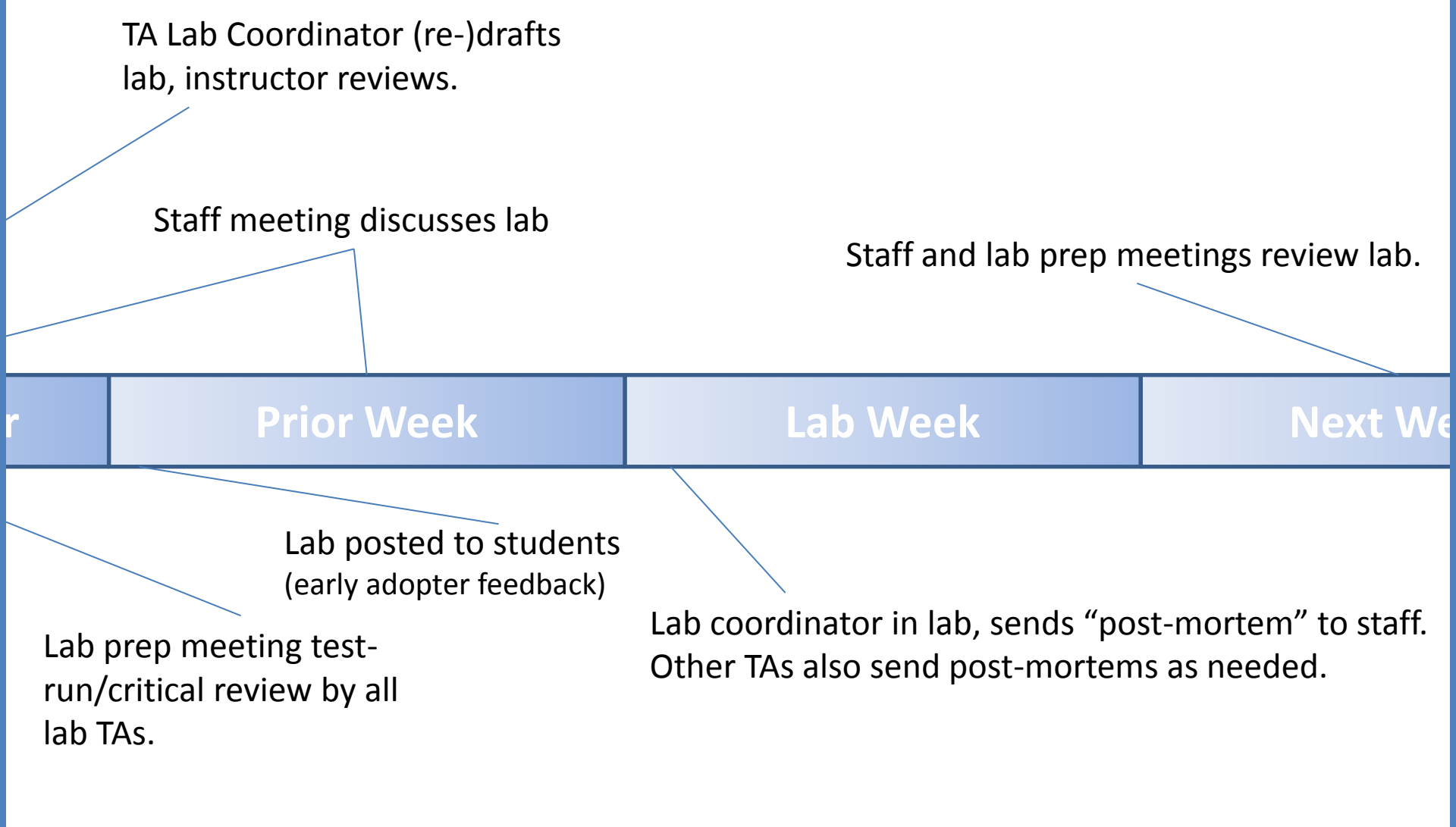
This year: developing an assessment to measure achievement of selected learning goals, rather than affective outcomes.

Design Process: Communication

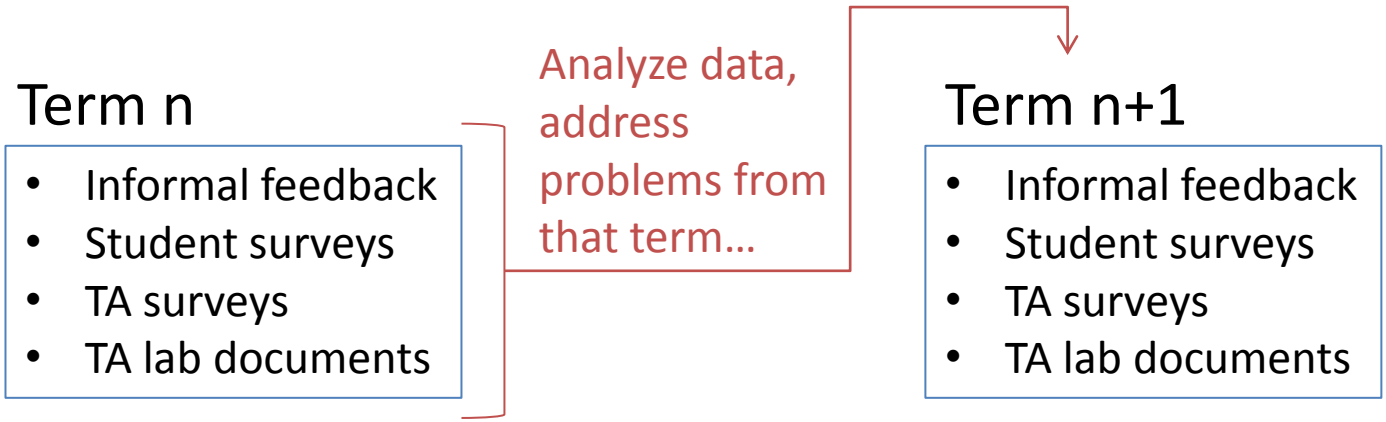


Formal appointment of a TA as "lab coordinator" enables broad, regular feedback from all stakeholders, feeding to improvements in labs.

Design Process: Weekly Timeline



Design Process: Term-to-Term Timeline



Changes recorded (and rendered sustainable) since "retirement" of second TA lab coordinator with "lab manager guide", "lab planner guide", and living "labby-lab" (lab by lab design document).

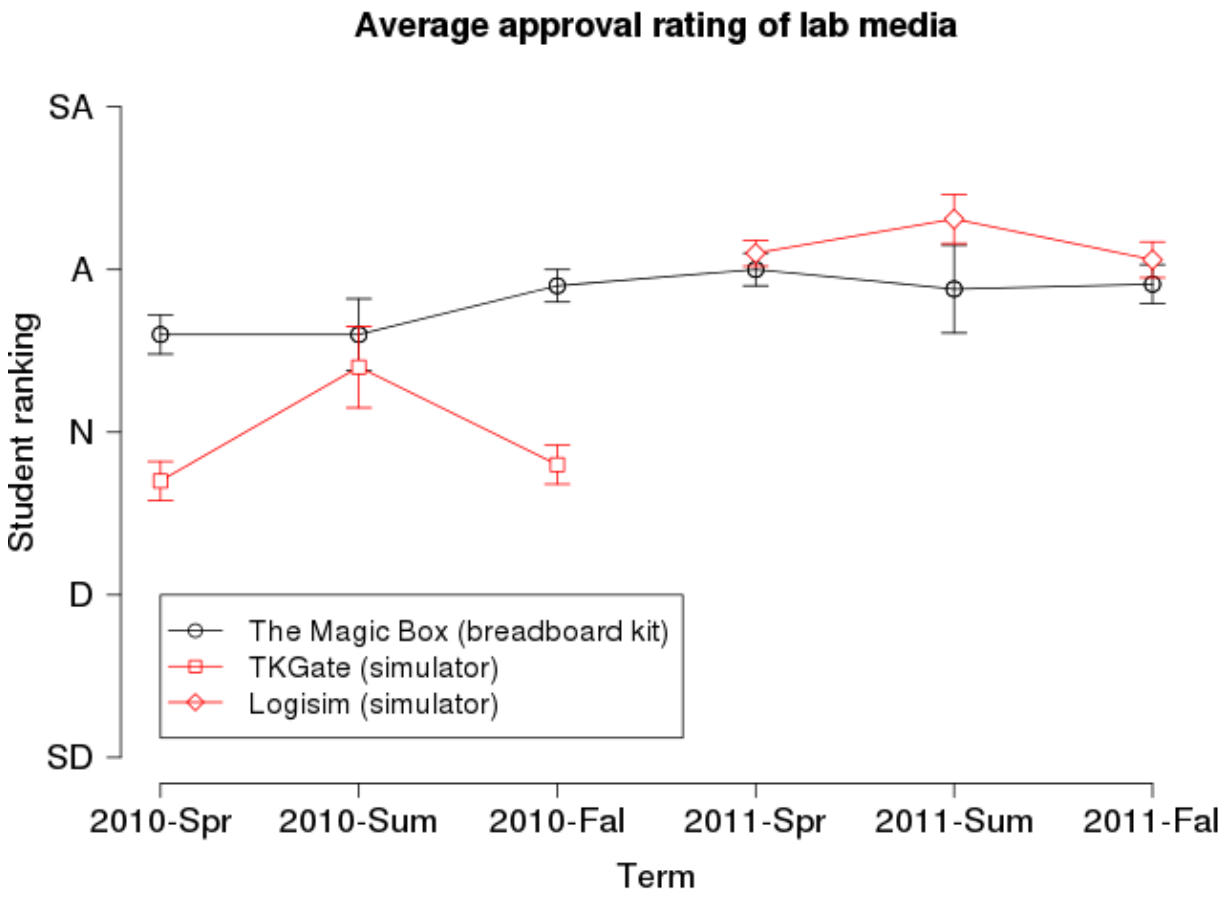
Example: TkGate/Logisim Change

In our first end of term student surveys qualitative and quantitative data agreed: *"tkgate sux!!!!!* [sic]*"*. (We also learnt that the crucial CPU lab was poorly received, and that the students loved the regular expressions lab.)

In the next two terms we tried improving our support of tkgate. Still no change in student feedback. "tkgate is totally not user friendly."

So we threw it out. We switched to logisim.

TkGate ⇒ Logisim



Qualitative feedback that term, and subsequent ones, didn't have complaints about the circuit simulator. ("no news is good news")

Quantitative feedback jumped up dramatically. (Compare to feedback on breadboarding kit, where only small term-to-term changes occurred.)

Other Examples

- Shorter lab documents
- (Ongoing) evidence-based TA redesign of instructor-designed marking scheme
- Break-up of “heavy” sequential circuitry lab
- Finally well-positioned first CPU lab
- Editing standards for labs

Acknowledgments: Head TAs (Rachel Busby, Mark Crowley, Ian Dewancker, Vanessa Kroeker), instructors (Meghan Allen, Patrice Belleville, Dave Tompkins, George Tskiknis, Kim Voll, Bob Woodham), other TAs (too many to cite!), and the many students who gave feedback! Plus Michele Ng, Anthony Winstanley, and Mark Greenstreet. Funding from CWSEI-CS and NSERC.