A New Twist on the Open vs. Closed Lab Debate: Is There an Advantage to Being a Laptop Campus?

James S. Bohy
Department of Computer Science
Simpson College
bohy@simpson.edu

Abstract

Recently, this researcher had the opportunity to teach a data structures course at two separate (but similar) institutions. Both classes used the same text/course materials, assignments, and examinations. The learners also faced a new (to them) course format. One of the schools had just gone to a closed-lab approach for their introductory curriculum, while the other had just adopted a laptop program for its computer science majors.

Students in each of the classes were asked to reflect weekly on their experiences; they were provided with prompts and responded via electronic mail. This researcher also reflected on his own perceptions of the effectiveness of each technique. Data on class activities (exam scores, programs, etc.) was also collected. The students also took the multiple choice section from the Advanced Placement Computer Science Exam (form AB). The results were inconclusive, but the idea does merit further exploration.

Introduction

Computer science education is a moving target. There are advocates for various programming languages (each with his/her claim of pedagogical superiority and/or practical relevance), programming paradigms (usually closely associated with language), and curriculum models (integrated approaches vs. non-integrated approaches). In each case, it is likely that there is not a universally applicable method that will work in every environment. Some schools likely survive on the ability to place their students with local employers and as such feel obligated to present more “practical” offerings; others can afford to take more risks (either because of academic reputation or due to a lack of local employment opportunities) and as a result can be more “theoretical.” Of course, we should look upon all such bandwagon issues with some level of skepticism (Kay, 1996).

One area in which there has been little (if any) debate in the past several years is over the format of labs for the introductory sequence. In the beginning, students worked on their programs outside of class time, but this was more of a practical issue more than it was a pedagogical choice. Computing resources were cost prohibitive and, as a result, scarce. Often times, there were not even adequate text materials available; this researcher has heard several anecdotes about "learning Pascal from the reference manual for the mainframe" over the years. With the introduction of terminals, personal computers, and workstations, the ability to do "lab" work in more of a classic science sense became a reality. With laptop computers and wireless networking, there is another revolutionary possibility. Portable carts with networked machines have been around for a while, but now schools are providing laptops to students as they arrive. The purpose of this study is to explore ways that computer science programs might take advantage of this.

The Problem

The two schools (hereafter referred to as School A and School B) that took part in this study are both small, private, church-affiliated liberal arts colleges located in the Midwest with fairly open admissions policies. School A offered degrees in computer science and computer information systems, School B offered a degree in computer science with several different tracks to choose from. School A had traditionally used an open lab approach for their introductory curriculum, where School B had used a closed lab setting.

When this study was conducted, each school had made a significant change. School A (where this researcher was a full-time faculty member) added a closed lab component to its CS1-CS2 sequence. School B had just started a laptop lease/purchase program for its incoming CS majors, and had altered the schedule of their 4-credit CS2 course from 3 hours of lecture with a 2-hour lab to 4 hours of lecture which should incorporate the laptops. Due to a shortage of instructors, this researcher picked up a section of School B's data structures course.

There are a number of confounding variables for which this researcher could not control. For example, School B's academic term started a week earlier than School A. This proved to be an advantage; while both schools were using C++ in their CS2 course, School B taught the CS1 course in Pascal. As such, the extra week was used to start the transition from Pascal to C++. The next week, both schools started with traditional data structures material, using Nell Dale's (1999) *C++ Plus Data Structures* as the text and teaching the course using an object-oriented approach (Fienup, 1996).

Methodology

While there are certainly results to share in terms of exam scores and programming ability, this is primarily a reflective piece. As such, the data analysis is almost entirely qualitative. The quantitative data is shared and its significance is discussed, but it is the qualitative part that carries the most weight. Because this piece is so reflective in nature, the voice of the paper will switch to first person from this point on, because that is deemed appropriate in qualitative literature (Krathwohl, 1993; Wolcott, 1988).

Not only were each of these settings new for the students at each institution, they were new for me as the instructor as well. As an undergraduate, I missed punched cards by a mere three years and did not do any programming on a personal computer until I was a senior. My early teaching experiences were also primarily open lab type courses, except for the occasional applications-oriented course that I would typically end up teaching in the lab. As such, I viewed the teaching of these two courses in two different settings as an opportunity not only to research, but to grow as an educator.

The notion of teacher as researcher is not a new one. The notion of *action research* "is based upon collaboration, participating, democratic decision making and emancipation through critical self reflection" (Kember, 1998, p. 54). Thus, my job was to tailor each of the courses around these principles. The primary way in this was accomplished was through an exercise of weekly reflection, not only by the students, but by myself as well. This process is described as the methodology unfolds.

Subjects

There were a total of 21 subjects in the study. 20 of the subjects were students enrolled in the two courses (8 at School A, 12 at School B). Only 3 of the 20 (15%) were female, and only 2 (10%) were of non-traditional age. There were 2 (10%) international students; these were the only minority students enrolled in either of the courses. I acted as the 21st subject, as my own reflections are woven in to the student reflections.

Data gathering

As previously mentioned, both qualitative and quantitative data were gathered for the study. This is an acceptable approach as the two methodologies can be used to complement each other as research and therapy complement each other in the field of psychology (Nau, 1995). The quantitative data consisted of the objective portions of two course examinations and 20 multiple choice questions from the Advanced Placement Computer Science (APCS) AB exam. The primary focus of the course examinations is on data structures concepts, where the APCS exam is almost entirely oriented toward the implementation of data structures concepts in C++.

The qualitative data from the study consists primarily of student and instructor reflections. For my part, I endeavored (not always successfully) to reflect on a regular basis on my experiences in the course. I tried to focus on how I was making the experiences different for the students, in particular how I made best use of the different course settings. More importantly, I tried very hard to make sure that my reflections included pieces from events that actually took place; my own interactions with students either in person or via e-mail.

The students were also asked to reflect periodically, though there was much more structure to their reflection exercises. The reflections were broken into two parts. The first part related directly to whatever topics were currently being covered in the course and was a series of questions. An example might be "what aspects of C++ (besides structs and classes) are you most comfortable with?" In this case, structs and classes

were intentionally left out as they had just been introduced that week. As should be obvious, this part of the journaling changed each time the assignment was given.
The second part of the reflection was a constant series of prompts. The students were asked to respond/react to at least three of the following:

> I was most engaged in learning when …
> I was least engaged in learning when …
> The positive thing about this class …
> I really wish we could …
> I found the [closed lab / laptop] to be useful when …
> The [closed lab / laptop] interfered when …
> In the future …

The reflections were primarily distributed via e-mail (Harris, 2002). The most important part of this exercise as an instructional tool was to make sure that it had an impact on the way I taught the classes (Brubacher, Case, & Reagan, 1994). In particular, it was key for me not to take the student comments personally (and to assure them that I would not). In an ideal research setting, I would have been able to get the student comments anonymously. The value of the comments from an instructional point of view would have been diminished, especially in cases where a particular student was struggling and needed more individual attention.

Results

The early reflections from the School B students definitely supported my own observations. The time spent in the first week comparing C++ to Pascal, and especially the web sites and other resources they were offered, really meant a lot to these students. In particular, one female student noted that the most positive thing about this class "…is that you're very helpful. The climate of the classroom is relaxed and open so we feel comfortable to ask questions in class or over email." This same student echoed a popular sentiment in noting that she found the laptops to be most useful when "…we used them in class. It's really convenient and useful to be able to try out code right then and there without going to the lab. This way, no one (well, almost no one) has to share a computer so almost everyone gets the hands-on experience that is very beneficial."

The School A students were using a closed lab setting for the first time; they all had CS1 using open labs. In their early reflections, they tended to gripe a bit about the structure of the lab. As we got more into the flow of the course, they recognized some positive aspects of the closed lab setting. For example, one male student noted that he found the labs to be useful when "… people talk and ask each other questions. This was useful because it was another way to learn the material rather than reading it from a book. More interaction verbally with each other on the material." Another male noted that the most positive thing about the class "…is that we have a lab time. It gives us a chance to be working on a program and have you and other classmates around to help." Having (essentially) dedicated access to the instructor was a real strong point.

As the course moves on toward the first examination, we are working with structs and classes. It turns out that some students at School B are starting to get a bit frustrated with the laptops. We don't use them every day, so there is the issue of having to carry them around. Also, some students have difficulty resisting the temptation to use them. I know that this is my "fault," in as much as I have still not figured out a way to seamlessly integrate them into the class. The School A students are getting used to the lab, but find that it is somewhat difficult to get the assignments done in the prescribed time.

From the outset, it seemed clear to me that the transition from Pascal to C++ for the students at School B was going to be my largest obstacle. In some sense, it was naïve to assume that I needed to run the two courses on the same schedule, but that was the way I chose to proceed. In my first reflection, which occurred after the fourth week of classes at School A, I observed that:

> After this first unit of the class, I would actually say that the [School B] people are just as well of with C++ as the [School A] people. I find that to be very revealing – it reinforces a notion that C++ may not be the most appropriate language to use for beginners. The [School B] folks are a bit stronger with programming concepts, simply because they've apparently been able to cover more of them using Pascal.
>
> But they are also highly frustrated with the way they are being taught C++. Learning language and concepts concurrently is always difficult. This problem is heightened when the concepts depend on a more thorough understanding of the language. I'm not sure what the answer is here – I don't believe that it is impossible to do [things] the way we're doing [them]. Reassurance may be the key, and perhaps expectations need to be made more clear.

In other words, the School B people apparently had a stronger background in terms of programming constructs than the School A people did, *even though the School B subjects tended to struggle more with the language due to the change*. The other thing that I have observed at this point in the course is that the events in lab at School A on Tuesday end up shaping what I do in class at School B for the rest of the week. I'm not entirely sure this is a positive development; I need to find some way to make assignments and activities fit. I have also lost people in both classes by this point.

The first exam is broken into an objective and a short answer portion. The worst score on the exam (60%) comes from School A. The rest of the scores are bunched pretty tightly around the mean for each school separately and for both schools overall. A t-test shows no significant difference between the two groups in terms of performance on this first exam. The overall average is about 79% (a high C). However, a look at the grade distribution for the two classes is somewhat more revealing, as seen in Figure 1.

While the standard deviations and means for the two schools are nearly equal, it is clear that the performance on the exam varied greatly between the two schools. Since this the primary focus of this examination is concepts related to data structures and not the coding itself, it is not clear that the course format is a determining factor in the differences seen here.

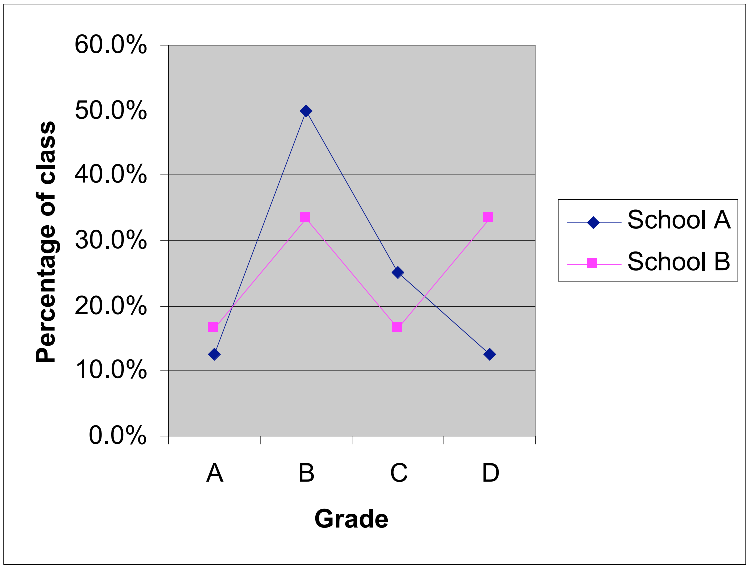

Figure 1: Grade distributions for the first exam for the two schools

As the course continues on, the unity in the comments from the students at the two schools begins to fade. Some continue to struggle with rudimentary things (like due dates for assignments). Others continue to react favorably to the methods being used. Some are starting to become rather cynical about the reflections, as one male student from School A put it when stating that the most positive thing about this class is that "…it meets twice a week." This is a step up from an earlier reflection where he indicated that the most positive thing about this class is "… I can't think of one."

I am starting to notice that the people at School B, who were initially handling the transition well, are now getting caught up in the language. Their own reflections show that. One male student notes that in the future "…[School B] should break this class into 2 classes, one class that just deal with Data Structures and the other class mainly for C++ because we are trying to learn about Data Structures but on the other hand we are trying to learn C++ now." He goes on to note that, "I think this class is going pretty good so far, at least for right now. I think as time progress with lab being due on time, you should be receiving more questions or emails."

The School A people, on the other hand, are starting to get into the routine of the lab. It is clear to me that they are moving on to the data structures concepts more ably than the School B folks. In my own reflections, I note that:

> One thing I haven't done a good job of with the [School B] people is to de-emphasize the language (C++). They have they idea that they must know every

> little detail regarding how it works before we can move on to other things. In looking at that, I need to approach this differently in the future if I am ever in another situation where language changes between CS1 and CS2.
>
> I am most surprised by the profound lack of questions from the [School B] people. They complain in their reflections (some of them, anyway), but they don't ever complain in class. They all seem a lot more personable than the [School A] people, who I know a lot better having had most of them in the class prior to this. The [School A] people just seem dead to me. They do seem to engage in the programs more during class than the [School B] people do. But again, I haven't made effective use of the laptops yet.

We're getting close to the end of the semester, and a colleague at School B has asked me to help pilot some questions on the APCS AB exam. I'm glad to do it, because it will be an external way for me to assess where the two classes are. I am allowed to grade the exams myself before returning them to Educational Testing Services (ETS). The standard is that each correct answer is worth 1 point, and each incorrect answer is -0.25 point penalty. Prior to giving the exam, I explain this system to both classes; that it is better to leave something blank (in terms of your overall score) if you are unsure.

The results of the APCS exam are much more telling than those on the first unit exam. This exam focuses more on C++ than my first exam did (the second unit exam and final exams do have coding on them, plus I give programming quizzes). We can break the data down in terms of the number of correct answers, the number of incorrect answers, the number left blank, and the final score. Table 1 shows these items for the given schools.

Table 1: Comparison of APCS exam results by School

	School A	School B
Number correct		
Mean:	4.25	5.00
Standard deviation:	1.28	1.28
Number incorrect		
Mean:	9.00	6.08
Standard deviation	1.85	3.12
Number left blank		
Mean:	6.88	9.22
Standard deviation	2.70	4.21
Scaled score		
Mean:	2.00	3.48
Standard deviation:	1.13	1.03

A t-test of unequal samples showed a difference between the two schools at the $p < 0.01$ level for both the scaled score and the number incorrect. In a sense, the students at School B took fewer chances, which is reflected in the fact that as a group they left a lot more blank (I'll assume the wider standard deviation accounts for the lack of significant difference on this particular mark).

As the semester winds down, the School B people are getting more and more frustrated with the way the laptops are being used. Part of it stems from not having a sufficient network connection in the classroom where we are meeting. We could really use a switch or a hub of some kind. Wireless or some sort of desktop docking station is clearly the way to go. My biggest frustration with the laptops in this configuration is that there is no easy way to share assignments. I do have access to a shared network drive, which is a nice repository. I continue to use my website as a place for them to get materials, but I find that even at this point in the term there are folks who haven't gone there yet!

The School A people are hard for me to get a read on. They seem to have a love-hate relationship with the closed lab setting. Some would like to go back to an open setting, others like the idea of always meeting in the lab and in essence doing what the folks at School B are doing. Use the computers when a teachable moment arises. I find that I am spending a lot of time trouble-shooting for folks in both classes. I'm so thankful for e-mail and internet access, especially since the schools are separated by an hour.

I haven't really formally reflected on my own progress since right before the AP thing. I am frustrated by the fact that neither group seems to have a handle on functions; a concept that is covered extensively in the CS1 course that both of them had. Each week I ask them about their comfort level with C++ or to tell me what areas in particular they are having difficulty with; neither group seems interested in being honest with me. A student will say he's a "9" (on a scale from 1 to 10) in terms of comfort with C++, but turn around and say he doesn't understand functions at all?

It's time for the final exam. Both schools show a dip in the mean score from the first exam, but at School A the drop is much more pronounced (an average of 79% on exam 1 to an average of 64% on the final). Three people at School A fail the final, which obviously contributes to the drop; two people fail it School B, but not nearly as badly as those at School A. The overall mean on the exam for both groups drops from 79% to 70%, and the standard deviation raises from 9.2 points to 12.8 points.

For my final reflection, I realize that two things have happened. One, I have not made nearly good enough use of the laptops at School B. Even though I observed this to be the case, I have not done anything about it. I look back at a previous reflection:

> This weekend, I need to take time to plan a little better. Plan seems like such an odd word for it, because what I'd like is for things to be a little more "free" in the [School B] setting. Do the laptop thing right.

At odds with this is the other thing that I have noticed. In spite of my inability to "do the laptop thing right," and in spite of their grumbling, the School B people have learned C++ and data structures concepts. In fact, judging by the quality of the programming assignments, they have passed up their School A counterparts in terms of the ability to write higher quality programs. I assure myself that their background in Pascal probably had more to do with that.

Conclusions

I learned a number of good lessons from this exercise. I offer my usual caveat that the mileage of the reader may vary. Reflection is difficult, for both students and instructors. It has to be done in a "safe" way; students need to feel free to be honest. If there is any notion of reprisal, then the reflection becomes a waste of time. In terms of the instructor's personal reflections, it's important not to get in the mode of complaining. The one thing I did not do (and in retrospect, wish I would have) was reflect back to the students. I responded to their reflections, but did not share mine with them. I think that sort of thing would have forced me to whine less and think more.

My first observation is that any change of class format has to be accompanied by an innovation in the way the class is conducted. The mere presence of the laptops should have allowed me to say, "well hey, let's see what happens when we do x," and then have them model it. Instead, I found myself saying, "be sure you bring your laptops to class on Friday, as we'll be doing a lab thing." This probably seems like an obvious observation, but I don't necessarily believe that. Networking would obviously have made a lot of what I thought I wanted to do easier.

The closed lab format is similar. I found myself trying to hard to make assignments "fit" in that 2-hour time slot. I still find it difficult to write 2-hour labs, though it's getting better. But I think that the point is that these small assignments in lab still need to be accompanied by larger assignments that they do outside of lab time. That was actually easier to conceive of with the laptop setting, because in that case, I only had an hour.

My second observation is that I'm not entirely sure what the "advantage" to the laptop format vs. closed-lab format is. Right now, it appears to just be different. The quantitative results were inconclusive; the only significant results came in the evaluation of the APCS exam, and frankly it could just be that the School B students were better at thinking, "you know what, I'm not sure about that, so I'm going to leave it alone." That being said, I think that pursuing the idea of bringing portable technologies into the classroom has merit for further explanation.

In particular, there would seem to be applications beyond the CS1-CS2 sequence. Using some form of portable technology to write protocol software in a networking class, or being able to share partial code or development tools in a software engineering class are just two examples. As I think about it now, the key here would have to be wireless networking. Wired networking, unless using docking stations, could get cumbersome.

There are still the same issues that exist in the lab setting, though. As was pointed out by a number of School B students, the temptation is there to play with the machines. Time spent waiting for them to start up is unacceptable; one really does need to have them start the machines the moment they enter the classroom. Without some program like SynchronEyes™, it is hard to know if they are engaged in learning or playing Internet poker during class (some may argue that this is learning, just not learning computer science).

My third observation is that I worried too much about keeping these classes in sync with each other. Yes, it was valuable in terms of being able to give the same exams and covering the same ground. However, it was also mostly impossible. The two courses were using different programming environments, and there were constant hardware and software problems. As noted previously, I found I couldn't just say, "okay, let's fire up the laptops and look at this." A lot of time wasted trying to do that.

In particular, my naïve approach to keeping the classes in sync was to use PowerPoint™. It is a wonderful tool for some things, but I found that I forgot everything I learned in my education courses in terms of overusing it and of packing too much information into it. I continue to beat presentation software to death in my teaching; re-reading my reflections has caused me to re-think my stance on it!

Which brings up my fourth and final observation; I think all faculty in computer science should engage in a semester of reflection like this. One does not need to ask for student feedback daily or even weekly. I was getting stuff from them every 3 weeks or so, and it was wonderful. However, the key is taking it seriously. I noticed that if I didn't respond to their comments in some way – by e-mail, in person, or in some observable change in what was going on in class – they took me to task. Worse yet, they often stopped taking the reflection seriously.

I have kept in touch with a number of these students. One of them got a job at School A after graduating from there, and I still see him from time to time (even though I no longer teach there). After telling me that the most positive thing about the class was that it met two days a week, I have noticed a different sort of attitude. At first, I thought it was just a product of the maturation process, but he harkened back to those days when we had that class and noted that he probably didn't appreciate what I was trying to do then, but he mostly understands it now. For me, that spoke volumes.

References

Brubacher, J. W., Case, C. W., & Reagan, T. G. (1994). *Becoming a Reflective Educator: How to Build a Culture of Inquiry in the Schools.* Thousand Oaks, CA: Corwin Press.

Dale, N. (1999). *C++ plus data structures.* Boston: Jones and Bartlett.

Fienup, M. (1996). Rethinking the CS-2 course with an object-oriented focus. *SIGCSE Bulletin (28),* 3, 23-25.

Harris, J. (2002). The correspondence method as a data-gathering technique in qualitative enquiry. *International Journal of Qualitative Methods(1),* 4, Article 1. Retrieved December 18, 2002 from http://www.ualberta.ca/~ijqm/

Kay, D. G. (1996). Bandwagons considered harmful, or the past as prologue in curriculum change. *SIGCSE Bulletin(28)*, 4, 55-58.

Kember, D. (1998). Action research: Towards an alternative framework for educational development. *Distance Education(19)*, 1, 43-63.

Krathwohl, D. R. (1993). *Methods of educational and social science research: An integrated approach.* White Plains, NY: Longman.

Nau, D. S. (1995). Mixing methodologies: Can bi-modal research be a valuable postpositivist tool? *The Qualitative Report(2)*, 3. Retrieved 12/18/02 from http://www.nova.edu/ssss/QR/QR2-3/nau.html

Wolcott, H. F. (1988). Ethnographic research in education. *Complementary Methods For Research in Education*, 187-206. Washington DC: American Educational Research Association.