Experience with In-person Grading

J. Philip East
Computer Science Department
University of Northern Iowa
east@cs.uni.edu

Abstract

During this past academic year I have added a new activity to my teaching repertoire—in-person or face-to-face grading. The activity involves a briefer than normal examination of assigned projects followed by meetings with individual students to examine the results of their work and to assess those results. This paper provides: a) an introductory discussion of personally held views about teaching and learning that serve as a rationale for this activity; b) a discussion of experience using this particular activity; and c) recommendations for those who might wish to try it themselves.

Introduction & Rationale

Readers who have encountered my past work at this conference probably recognize that I, and my teaching, are works-in-progress. In the past, I might have considered that to be negative but I now believe it to be positive. As I teach, I occasionally reflect on my practice and then consider changes that might make my instruction better. While doing this I also work to enhance and explicate my theories of learning and of teaching and the connection between the two.

A rather rough description of my perspective on learning is that we (people) exist within contexts and learn in relation to those contexts. At any moment in time we perceive much of the current context via our senses of hearing, sight, touch, etc. However, we attend to only some of what we perceive. Our current state of mind and body affects what we perceive and attend to. After perception occurs, our mind interprets the perceptions and connects current experience to past experience. Again, our current state of mind and body will affect the interpretation and our mind's connecting it to past experience. Future knowledge or understanding then depends on what an individual perceived and attended to and the manner in which it became connected to extremely individual experience.

So, what might this view of *learning* say about *teaching*? Perhaps the most important implication is that we should recognize that each of our students brings a unique understanding or misunderstanding or lack of understanding to whatever we wish them to learn. They have all had different experiences and we cannot assume they are the same. Also, as we teach, students may not perceive or attend to all we say and do—they may be thinking of their boy/girlfriends, fighting a headache, dozing, etc. Even those students who do perceive and attend to what we say and do may actually learn something different than we expected. This could happen because students' minds still have to interpret the perceptions and connect them to their individual past experience. Clearly, we teachers will have very little insight into what individual students learn from us unless we can somehow see into their heads.

A second implication of this view of learning suggests that our teaching activity ought to be designed to more actively involve the students. We should reduce the number of opportunities to think about boy/girlfriends or doze, i.e., we should depend less on telling or lecture which allows their minds to more freely engage in non-productive activity.

In planning instruction, I see two levels at which we might reflect on our practice. We can consider what happens in our classes on a daily basis, i.e., how did it go today or how might I improve my performance in this particular teaching activity? Alternatively, we can reflect at a more global level by considering the activities we have employed in our courses and how we might do something more useful. I assume some of us will be more comfortable with one type of reflection than with the other. I personally, seem more comfortable with and interested in the course-level reflection. This paper discusses an activity new to my course plans—in-person grading.

Last year there was some discussion on the SIGCSE[*] e-mail discussion list about in-person grading. A reference was made to Cooper's teacher's manual for *Oh! Pascal!* [1] and its discussion of face-to-face grading. I found and examined Cooper's ideas and decided they were worth considering. His claim, "it is probably the single most important improvement that can be made in course management" (p.130) was quite compelling.

Additionally, for some time my wife (who teaches child development) has been meeting with her students several times a semester to discuss their work. In particular, at the end of the semester she meets with the students to discuss with them a self-evaluation of performance in her course. I do have students evaluate their computer programs [2], but had never really felt comfortable meeting and discussing individual performance with students. Note, however, that in this case, my teaching behavior is at odds with my strong teaching belief that I needed to get inside students' heads to help them learn.

My brain eventually made a connection between these two bits of perception—that my theory/model of teaching demands I strive to see what is in student heads but that I refuse to do in any planned way. I started considering whether and how I might change my teaching to reap the supposed benefits of in-person grading.

In the rest of this paper, I describe my in-person grading activities and my reactions to them. I close with some thoughts and recommendations the reader may want to consider. I have one other comment before moving to those topics, however. If you, like me, wish to reflect on and try to improve your teaching, please take care. I strongly suggest you make only minor and, preferably, single changes in your instructional practice at one time. Doing so will allow you to focus on the one change rather than dividing your attention among several. Additionally, you will be better able to assess the results if only one major variable has changed. Finally, if for you as for me, change is hard, perhaps one small change is easier than wholesale change.

My In-person Grading Activity

By the fall of 2000, I had determined to use in-person grading in my *Design and Analysis of Algorithms* class. My goals for this course include:

- true understanding of the implications of algorithm complexity, i.e., students will analyze problem solutions for time and space complexity and will incorporate the results of the analysis into design decisions
- examination of the importance of and alternatives for data representation in algorithms, i.e., students will expand their repertoire of data representation techniques and will make considered decisions about problem and data representation
- consideration of alternative design approaches, i.e., students will be able to choose and use appropriate design techniques such as brute force, greedy, divide and conquer, dynamic programming, and heuristic search

[*] Special Interest Group on Computer Science Education—an affiliate/unit of the ACM (Association for Computing Machinery)

- program assessment, i.e., students will (relatively accurately) assess the quality of their programs with respect to correctness, design, documentation, code layout, etc.

I had taught the course two or three times prior to that semester and was not very happy with my performance or the students' performance. Texts for this course tended to be of two types. One kind of text is designed for a relatively theory-oriented course that both undergraduates and graduates might take. The other kind of text addressed data structures and introduced the "algorithms" topics. Neither kind of text really did what I wished to do. Additionally, I was not familiar enough with the material to strike out on my own. My instructional approach involved:

- students reading the portions of texts that I thought were appropriate (there was quite a bit of skipping around)
- my presenting problems relating to the current topics and using class-time to discuss them
- students working on projects and programs relating to the topics
- my grading and commenting on the assignments and (eventually) returning them to the students

I enjoy problem-oriented class discussion and am usually able to make a variety of points about problems and algorithm design. Generally speaking the class went fairly well even though I stumbled on occasion in the class discussions. However, students were not happy and, more importantly, they didn't seem to be learning what I thought was important. Clearly, something needed to change.

I was particularly disturbed by the fact that I had spent considerable time and effort addressing time complexity but the students did not perform well on exam items related to it. In the fall of 2000, I was determined to have a better outcome. I announced to my Algorithms students that I would do some in-person grading.

In the fall semester I met with students individually on four occasions. The first related to a project in which they were to create an artifact that illustrated their understanding of complexity. The other meetings considered student work on programming projects.

The complexity project was rather loosely defined. Students were to demonstrate that they understood complexity. They might write a paper, produce a spreadsheet, provide and discuss program segments, etc. that would convince me that they had grasped the concept of time complexity. They were to turn their work in by a particular Friday and sign up for an in-person grading appointment to occur on Monday, Wednesday, or Friday of the next week.

Over the weekend I examined the projects about which we would meet on Monday (and similarly, on Tuesday evening I examined work with Wednesday appointments and on Thursday evening I examined work for the remaining students). I prepared some questions to ask students about their work. Some of the questions related to the intent or correctness of their work and some addressed the students' general understanding. I also asked each student to rate their understanding of complexity—poor, okay, good, or excellent. The meetings were 12 minutes long and took most of two days to complete (about 25 meetings). Examining the projects before the meetings required about 10 minutes per project.

As might be expected I was a bit nervous in the meetings. I imagine the students were also. No meeting went particularly poorly and many were quite interesting. A number of the interactions turned into explanatory discussions during which I tried to clarify student understanding. Two of the students wanted opted to try to revise their work in an effort to better understand complexity. In this meeting I did not share my assessment of (grade for) the students' work.

The other applications of this technique in that semester related to programming projects in which students were to implement algorithms we had discussed in class. Again, appointments were made and the students work discussed. In these cases, I suggested but did not require that students assess the quality of their programs, perhaps by referring to a guide to program quality that I had developed (see Figure 1).

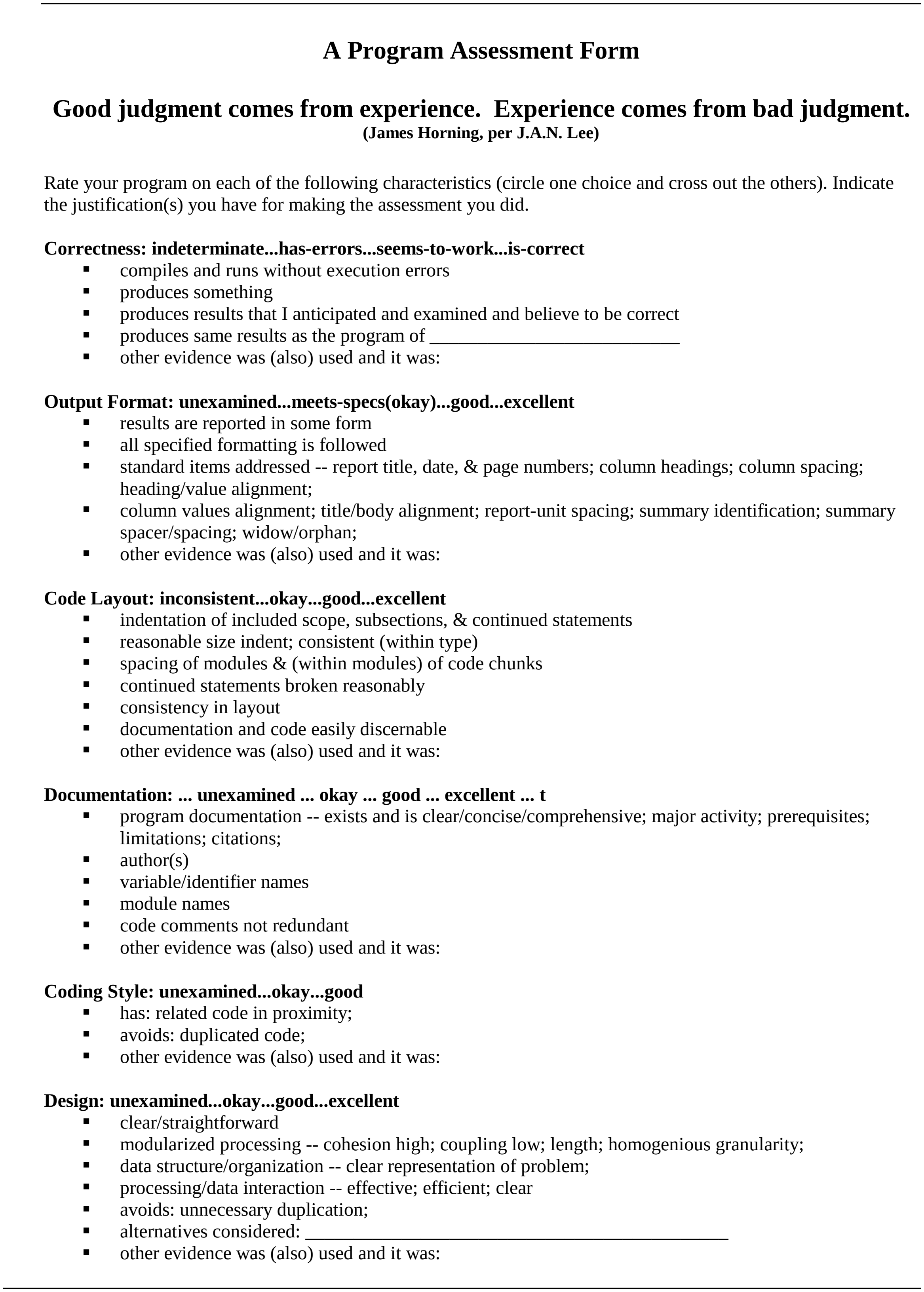
As with the non-programming project I examined student submission in advance of our discussions. This time my preparation consisted of either preparing questions, marking parts of the program that I considered to be good or in need of further attention. In some cases, I allowed 20 minutes for interactions rather than 12.

I noticed a number of things during this activity. I had hoped that I might spend about the same amount of time (or even less time) in this grading process than I had with the traditional process of carefully examining the projects and providing substantial feedback. Of course that did not happen—I used more time, perhaps 50 percent more. However, instead of feeling like I had to work extremely hard to be consistent in my grading, I found I could examine the projects rather informally (while watching television) since I would have another opportunity to see them during discussions with students. Thus, while more time was required it was less intense time.

A second observation took me by surprise. Students would often consider a program to be "okay" or even "good" when it obviously did not produce the desired or correct results. In these cases I usually asked if the student really believed a program that obviously did not work correctly was "okay". Additionally, I would note that we were assessing the program, not the student. The student could be okay or good even if the program was abysmal.

I was somewhat surprised that the additional time required did not seem burdensome. I did not have to find a big block of time to do the grading in an effort to maximize consistency. Also, at the end of the week, I was done! It has been years since I finished grading all class projects within a week of their submission.

The most significant observation was my reaction to the in-person discussion. Even though I am very uncomfortable when pointing out deficiencies in work, these meetings were mostly enjoyable. Actually, the most enjoyable ones were those with students



A Program Assessment Form

Good judgment comes from experience. Experience comes from bad judgment.
(James Horning, per J.A.N. Lee)

Rate your program on each of the following characteristics (circle one choice and cross out the others). Indicate the justification(s) you have for making the assessment you did.

Correctness: indeterminate...has-errors...seems-to-work...is-correct

- compiles and runs without execution errors
- produces something
- produces results that I anticipated and examined and believe to be correct
- produces same results as the program of ___________________________
- other evidence was (also) used and it was:

Output Format: unexamined...meets-specs(okay)...good...excellent

- results are reported in some form
- all specified formatting is followed
- standard items addressed -- report title, date, & page numbers; column headings; column spacing; heading/value alignment;
- column values alignment; title/body alignment; report-unit spacing; summary identification; summary spacer/spacing; widow/orphan;
- other evidence was (also) used and it was:

Code Layout: inconsistent...okay...good...excellent

- indentation of included scope, subsections, & continued statements
- reasonable size indent; consistent (within type)
- spacing of modules & (within modules) of code chunks
- continued statements broken reasonably
- consistency in layout
- documentation and code easily discernable
- other evidence was (also) used and it was:

Documentation: ... unexamined ... okay ... good ... excellent ... t

- program documentation -- exists and is clear/concise/comprehensive; major activity; prerequisites; limitations; citations;
- author(s)
- variable/identifier names
- module names
- code comments not redundant
- other evidence was (also) used and it was:

Coding Style: unexamined...okay...good

- has: related code in proximity;
- avoids: duplicated code;
- other evidence was (also) used and it was:

Design: unexamined...okay...good...excellent

- clear/straightforward
- modularized processing -- cohesion high; coupling low; length; homogenious granularity;
- data structure/organization -- clear representation of problem;
- processing/data interaction -- effective; efficient; clear
- avoids: unnecessary duplication;
- alternatives considered: ____
- other evidence was (also) used and it was:

Figure 1. Guide to Program Quality Used in Fall 2000

who had tried but not produced very high-quality work. In these cases the discussions became opportunities to teach. The activity was extremely satisfying. I felt that I was truly teaching and not just reciting the content or demonstrating my own abilities as a programmer.

This semester I am also using in-person grading, but some procedural alterations have been made. This time, I used a programming project to attempt to have students recognize the implications of algorithm complexity. Also, instead of just examining the printed program results I had the students compile and run their programs for me. Each student submitted the program code along with an explicit assessment of the program. I had created the assessment form they were to use (see Figure 2) and we had some discussion in class as to the qualities of a good program. At the time of this writing I have used in-person grading a second time following essentially the same procedures as with the first. I expect to do so one additional time, perhaps allowing the students to choose traditional grading or in-person grading.

Program Assessment of/by ____
(printed name)

Aspect of Program Quality	**Assessment of Component's Quality** (abysmal poor okay good excellent)
specifications are addressed (assigned tasks, assigned techniques, ...)	I-------I-------I-------I-------I
performs tasks correctly	I-------I-------I-------I-------I
algorithm is well designed (modularity, cohesion, coupling, consistent granularity)	I-------I-------I-------I-------I
algorithm is well implemented (straightforward, non-repetitive, naming conventions, ...)	I-------I-------I-------I-------I
code is well laid out (spacing, indentation, line continuations, ...)	I-------I-------I-------I-------I
code is well documented (modules, names, comments do not duplicate & are distinguishable from code, ...)	I-------I-------I-------I-------I
output/report format is appropriate (consistent spacing, concise non-repetitive description, ...)	I-------I-------I-------I-------I
submission content and format is appropriate ("portrait", item order, two-up?, ...)	I-------I-------I-------I-------I
was done on time (coding & testing, submission, discussion?)	I-------I-------I-------I-------I
submitter(s) completed & understand(s) the entire work.	I-------I-------I-------I-------I

I performed the work attached and assessed above. I am prepared to argue the validity of the assessment based on a standard of performance/quality (not based on: time spent, effort expended, my value as a human being, etc.)

____ ____________________
(signature) (date)

Copyright © MMI by Philip East, all rights reserved

Figure 2. Program Quality Rubric Used in Spring 2001

When examining the first project programs, I tried to consistently and carefully examine program layout and documentation. I did so in hopes that I might avoid having to address these issues later. Of course, I also examined the programs for correctness and design noting examples of both good and poor practice. Instead of 12 or 20 minute appointments this first program required 30 minutes (in addition to the 10-15 minutes of grading time before meeting). Actual discussions with students were quite similar to those in the earlier semester. I was becoming more comfortable with the process.

I try to end all the meetings with a review of the assessment rubric. The students have already indicated their assessments and during my pre-conference review I mark mine. I use the quality descriptors—abysmal, poor, okay, good, excellent—in my judgment and encourage students to do the same. At the end of our discussion I note areas of agreement and disagreement. The final act is the recording of a grade. We equate the descriptors with numbers (e.g., 0, 3, 6, 9, 12) and I ask the students to indicate the overall quality of the program with a number corresponding to some point on the continuum from abysmal to excellent. I then decide what number I will actually place on the work. After the first meeting, we typically are quite close in our estimations.

This semester, I had a few more students in the class, more of them submitted work and kept appointments, and the retention rate in the class was higher. The weeks during which I scheduled the grading appointments were very hectic and tiring. However, I still plan to use this approach in the future. I feel better doing it so it is worth the extra time. If the students benefit, so much the better.

I have made a number of observations beyond those noted earlier

The informal discussion of my expectations regarding layout, documentation, etc. made *me* feel these expectations were less arbitrary. It also seemed that students paid more attention to my suggestions for improvement. Actually, that feeling was true for nearly all the comments I made about their programs (not just documentation and layout).

There are always students who actually get almost nothing useful done on their programs. With in-person grading, however, several students have opted to revise their programs. I am sure that in some cases they wish to "get more points" and I point out that working on past assignments often reduces scores on current work. While some students are only after points, it seems certain that other students are genuinely interested in redoing the work to enhance their learning. This may not be due to the in-person grading per se, but the opportunity would not have (had not) arisen in my "normal" teaching practice.

Having discussions with students about their performance, rather than just grading it, also seems to allow me to focus on learning and the improvement of student skill or practice. For example, when faced with criticism of their programs, a number of students seem to engage rather automatically in rationalizing their performance. After multiple sets of meetings with students I noticed that I would tend to cut short these efforts. My comments on their work were not about the students' intelligence or worth but rather an effort to get them to consider alternatives that might well be better. And, I told them that.

Again, I do not know if students actually considered the comments in the manner I would prefer. I do know, however, that my previous teaching practice would not have allowed my making this point with any hope of not sounding patronizing.

I am still somewhat uncomfortable in the different role I need to play when meeting with the students in an evaluative context. In the classroom, I am the expert and in authority (even when I make mistakes). While I still possess the ultimate authority over grades, that role is not appropriate for the grading meetings. The primary purpose of those meetings is to influence student performance, not give grades. A secondary purpose is to hone their ability to judge their own work. My expertise and fluency with inter-personal relationships is not well developed enough for me to discuss this point with authority. I am sure, however, that the tone of the meetings cannot be authoritative or communicate negative views about the student if they (the meetings) are to accomplish my goals.

Conclusions and Recommendations

It is clear to me that I feel better about myself and my teaching practice as a result of using in-person grading. It allows me to (feel that I) focus on learning rather than on grading even though it is a grading process. An analogy may be useful. Back in the old days we taught programming in BASIC and FORTRAN and attempted to teach students to follow good practice. Pascal eventually became the near-universal teaching language because it not only allowed for good practice but also encouraged it. To me, in-person grading encourages better teaching practice and better learning practice.

Additionally, it allows me an opportunity to get inside the students' heads, thus making my teaching practice agree more with my beliefs about learning and teaching. I recommend it highly to all teachers.

I have some suggestions for those who want to consider including in-person grading in their teaching practice.

- Plan to spend more time grading. I had naively thought I might make this change and reduce or at least not expand the time I spent on grading. Instead the time has increased. But, it is less intense time and I am coming to actually enjoy the meetings with the students.
- Remember that one major reason for doing in-person grading is to get inside the students' heads to determine whether and what they understand and can do. Use questions or invitations so the students will talk rather than reporting the flaws you found.
- If you are not already comfortable with student meetings of this sort, do some role-playing or other thinking about the meetings and how you wish them to proceed.
- Use in-person grading on larger projects rather than daily or weekly homework unless the understanding desired is critical for future progress.
- Use pre-meeting examinations of work to prepare for the meeting. Highlight the questions you have or the points you wish to make.

- If possible, read Cooper's [1] teaching manual, particularly his suggestions for face-to-face grading and student interaction (p. 111-113, 116-119, 129-131).

Of particular note from Cooper's work, in my opinion, are:

- Consider evaluating programs as belonging to one of three categories: "great, all right, and awful" (p.111). (I might substitute "pretty darn good" for great.) Then, in discussions, indicate the assessment and delve into evidence supporting the assessment (rather than providing a laundry list of flaws). Doing so might reduce grading time and almost certainly will allow you to focus more on overall program quality.
- "Always, *always* refer to 'the program' rather than to 'your program.' Avoid putting the student o the defensive—a student who is busy defending her work isn't listening to suggestions for improvement next time." (p.113)
- Make criticism effective. Keep the list of shortcomings short and include good points (every program has some good points).
- Recognize that "poor programmers get the least inherent reward from their work. They need the most encouragement to go back and try to do better on the next assignment." (p. 113)
- Give the student a goal. After identifying some difficulties, suggest the student pay particular attention to one or two items the next time.
- Ask questions.

It seems to me that our ultimate goal as teachers is that all students develop the skill and understanding necessary to perform as computing professionals. If we are to be successful, we must deal with students individually to determine the extent of their skill and understanding. That requires meeting and working with them in contexts such as in-person grading. I wish you well as you consider how you can better work toward more successful student learning.

References

1. Cooper, D. (1995). *Teaching Introductory Programming (with Oh! Pascal! Second Edition)*. New York: W.W. Norton & Company.
2. East, P. (2000). Teaching, Assigning, and Grading: Continuing Efforts to Enhance Learning. Proceedings of the 33nd Midwest Instruction and Computing Symposium (formerly the Small College Computing Symposium, on CD-ROM), April 13-15, 2000. St. Paul, Minnesota.

Acknowledgements

I would like to acknowledge my wife, Katheryn Kinnard East. She has greatly influenced my thinking about and practice of teaching. Much of what I do in teaching is either borrowed from her or discussed with her before its use.