Providing Feedback to CS Students: Some Alternatives

J. Philip East
Computer Science Department
University of Northern Iowa
Cedar Falls, IA 50614-0506
east@cs.uni.edu

This paper describes some of my recent attempts to incorporate into my teaching activities that enhance the provision of student feedback. Ultimately, that means I actually grade less, since the biggest impediment to my providing quality feedback was that I didn't get the grading done quickly. I first provide some personal history and a rationale for these efforts. Then, I describe several of them. Finally, there is a short closing.

Background & Rationale

Grading student work is a discouraging task. A number a reasons contribute to make it so. First, providing good feedback on student work takes a lot of time. Furthermore, it appears that students attend mostly to the score on the assignment. They seem to disregard the advice provided in the feedback. This may be due in part to the fact that the feedback is not timely. By the time I make the comments, the instructional topic has changed and the feedback seems not to be relevant (though of course it is).

Thus, there is a problem. I spend considerable time on an activity that I believe to be useless. A rational person would not do this. How did I get to this point?

My teaching behavior and philosophy has changed significantly over time. I started out teaching as I was taught (or at least as I thought I was taught), by telling and perhaps showing students the things I thought they needed to learn. I organized the content in what appeared to me as a logical sequence and then I presented, or told, it to the students. As I was teaching math and programming, the teaching often included demonstration. I would then assign work that would allow students to practice doing what I had so clearly presented. They would complete the assignment and I would grade and return their work. Better teaching meant organizing the content in a better way or choosing better examples or explaining the material more clearly.

I eventually began work in a permanent position where part of my job was to teach (both preservice and inservice) high school teachers about teaching computer science. I soon discovered I had nothing really coherent to present to my students about teaching. I began a serious and long-term examination of teaching. For me that meant a search for a rational model of teaching and learning. I found no rational basis for in I had been doing. To some extent I threw the baby out with the bathwater. While I do believe my students learned, there was a period during which I now feel my teaching was not particularly good. I was floundering because I no longer believed what I had been doing had any rational basis but I had yet to discover something that did. I also noticed that students often seemed not to pay attention to the notes I put on their assignments as I graded them.

A number of realizations eventually led to some reasonably coherent picture of what I think constitutes good teaching. First, I noticed that even though I might explain something clearly and repeatedly, many students seemed not to "learn" it. I also remembered the saying that people remember only 10% of what they hear (or was it 15% or 20%). If students actually remember 20% of what I say, and I say ten things in a class I'd like them to remember, each will remember

only two things. To make matters much worse, it is likely that no two students remember the same two things. It also became clear that students might actually hear something different from what I said. Regardless of how well I organize and present information, students will haphazardly remember only a small part of it! Furthermore, I have no idea what any particular student might know.

Thus, I came to see that, much as computer science is not the same as programming, teaching is not the same as telling. So, if what I have to say and how I say it is not important, what is? Eventually, I came to believe that I should focus, not on what I want the students to know, but rather, on what I want them to **do**.

Changing teaching from telling to having students perform, requires that I reconsider the feedback I provide students[1]. If I want students to be able to correctly trace programs, I must provide them practice doing that. If the goal is that students be able to test their own programs, they must have practice in developing test data and testing their programs, not just running programs on my test data or knowing what "white-box" and "black-box" testing mean. If I want students to write good programs, I must not only inform them of the characteristics of good programs, but must also have them actually evaluate programs so they will have know what good programs are and when they have written one.

As I begin relying more heavily on having students do things in order to learn, the role of feedback becomes much more critical. Students must be informed when their performance needs improvement and where the improvement should occur. It is also very important that the feedback occur in a timely fashion. My past grading practices clearly do not meet both these goals. I could provide good feedback or timely feedback, but not feedback that was both good and timely.

It became very clear that I would not be able to consistently grade papers in a way the provided good feedback quickly. I might be able to get one set of work back to the students quickly, but not every set or even most of them. If I were going to change something, the change would have to include both **teaching** and **grading**, not just grading.

How can I provide better teaching that addresses these tasks and provides feedback to students on their progress? In the rest of the paper, I describe some activities that I have tried in order to accomplish this. I believe the activities themselves are useful, but the recognition of the need for them and for a continual search for ways to improve student learning is more important.

Merging Teaching and Grading

I began trying to adjust my teaching or in-class activity to make it easier to provide feedback that is reasonable both in quality and timeliness. The more I reflect on the notion of providing better feedback, the more important I believe it to be. Thus, I am seeking to make class activity include as little "telling" as I can and as much practice and feedback as possible.

I also am trying to get the students more involved in providing feedback to peers. Thus, I am also making significantly more use of group work. Eventually, I hope this enhances their ability to monitor the quality of their own work, particularly, when I throw in a little direct practice on doing so.

---

1 Actually, in computer science, the kind of feedback we try to provide is probably appropriate to our goals but not our teaching. We often teach language syntax and semantics but provide feedback on program correctness and quality. The difficulty is that we don't actually "teach" correctness and quality very much.

While I have performed no formal study to determine the efficacy of these approaches, they provide, for me, a rational model for examining my teaching. Also, I just feel better about my teaching. My teaching is much more fun for me now.

Critiques of Student Work

The first thing I thought to do was to discuss student programs in class. I offered a bit of extra credit for students who e-mailed me the code for their programs. At first, the discussion occurred immediately after the due date. Later, I moved it up and the discussion occurred prior to the due date providing students who desired to do so the opportunity to revise their programs (to actually practice whatever they took away from the feedback session).

I was actually surprised by the amount of participation. A third to a half of the class submitted code. There were plenty of submissions for me to find examples of good, less good, and unusual algorithmic approaches, and good and less good code layout and documentation. Some of the chosen samples were relatively large and some were small.

Students participated quite well in the discussion and appeared genuinely interested. I felt comfortable discussing the "good" examples. The "unusual" one were also relatively easy—I merely indicated that I would never have thought of that approach. The result would be okay so long as it was easy to follow. I worried that I would embarrass students with the "less good" samples. As I had given them some design and coding guidelines I was able to compare samples with the guidelines and discuss specifically how they differed. In many cases I also tried to place myself in the student's head and could voice a rationale for even the less good code. In any case, even the discussion of the less good examples went well.

Discussing or critiquing samples of student work seems very similar to examining case studies. The source of the material, however, is from the students rather than the instructor and the discussion is less indepth than in a case study. I believe seeing many examples of good and poor work allows students to more quickly develop a sense for good quality. The fact that the students themselves contribute the examples seems more powerful than if I were to generate them myself or even use examples from past classes.

I use student work critiques only for feedback. There is no grading component (other than possible awarding of a minute amount of extra credit). However, since program quality issues have already been addressed in class discussion, I do not have to include it when I check the programs. The grading is much quicker as all I have to do is check for correctness and note a score for quality. While I have had some difficulty breaking the habit of fairly extensive comments about program quality, I should not need to comment on quality as we have already addressed it.

This kind of activity should be useful for programs or for any other **design** products. Thus, instruction for requirements specification, initial design, algorithm development, interface design, test data development, and coding could all make use of critiques of student work.

Board Work

I have also been teaching discrete structures recently. While the student critique of student work can be applied to proofs and some other complex activities, much of the content in discrete is short-answer oriented. Additionally, I am less concerned about developing high levels of skill at mathematical proofs. Some other activity to enhance feedback is needed.

At first, I tried merely working the problems and having students compare their work to mine. It didn't work well. Too many of them either didn't do the assigned work or very superficially checked their work. Then I recalled my high school days of putting math homework on the board.

It connected with an idea from a colleague, John McCormick. John picks students at random to respond to questions on the assigned reading/homework.

I still provide some overview and review discussion that is similar to my past teaching. I no longer tell the students what is in the book, however. The bulk of class activity now consists of responding to student questions and checking work. At least that is the case when the homework consists of relatively simple exercises.

Students are given an assignment. There is at least one opportunity to ask questions prior to checking the work. Then we do the "grading". I have generated multiple randomized class rosters. I merely go down the list calling on the next student for the next exercise. There isn't actually enough chalkboard space to have the students go to the board so, initially, I handed out transparency foil and overhead marker to each student being called on. They simultaneously copied their work to the transparency. Then we check the results. More recently, I have begun to have them recite their answers orally while I write them down on the transparency. On each problem there is an opportunity for questions and comments and students will usually catch any error. In the early stages of this activity, I strongly encourage the students to ask questions, noting that if their solution looks different it might actually be different (and incorrect) and that they would miss a similar question on the quiz.

This allows for immediate feedback to the reciting student and to any others who wish to receive it. There is also time for discussion on each exercise. The students seem to be actually checking to see if their work is correct.

For the grading part of this activity, each student called on either: was basically correct, was basically incorrect, had no solution, or was absent. I am still experimenting a bit with the grading aspects but I currently am giving scores of 2, 1, or 0 or noting that the student was absent. By being absent a student misses the opportunity to recite. This part of the homework grade will be the average of all the recitation scores. (Thank goodness there are spreadsheets.) In addition to the homework scored from this grading there are chapter quizzes. Thus, the students who do not do the work will have ample opportunity to hurt their grade.

I try to avoid hurrying through the checking activity and to encourage discussion and questions. I also find this to be an excellent opportunity to emphasize points in context. Previously, I would lecture on important points and generally felt students didn't really hear what I was saying. Now, it seems to me that there is a better chance that they will actually attend to my comments.

General Group Work

I had the students form groups and then sit so the groups would be able to talk among themselves. I told them that before each class period they should be checking with their group members to see if they have a common understanding of the exercises. If they had any difficulty they should bring it up during this before-class discussion. If it wasn't resolved in the group, they should ask a question in class. They would then have at least some opportunity to work the problem before we checked their work. Some of the groups have extended their cooperation to a study group that meets outside the class.

Group Assignments

I also specifically use group assignments on more complex activity. While I do use some group "research" reports in my computer literacy class, the discussion here refers mainly to group assignments on proof-like problems in discrete structures.

Students all work, or attempt, the assigned problems and bring them to class. After an opportunity to ask questions the groups are asked to produce a group answer to a few of the problems. I suggest that the student with the least confidence on a particular problem be the scribe for it. Some class time is available for the group work but outside-class work also occurs.

The proofs are submitted to me for grading. However, there is only one-third as much grading and the number of items submitted may be fewer than assigned. Thus, I am almost always able to have quick turnaround on the grading. Group work, when done well, allows students the opportunity to teach each other and to provide feedback as to correctness and completeness. While I don't believe the work is done well by all groups all the time, I do believe the students who are actually interested in developing their skill do have a better opportunity than they might if I used traditional instruction and feedback.

I grade the group assignments much as I would any other proof (or program or whatever was submitted). The group members initially receive identical grades. They do, however, have the opportunity to evaluate each other's contributions to the group. I ask them to report on the participation of each member of the group by submitting a group report form (see Figure 1). I have used variations of the form in different classes. For large assignments (e.g., papers and projects) a form is turned in for each assignment. For a series of small assignments I might just have the students submit a single form at the end of the term or have them submit a form several times during the term. I average the percentages reported and use it as a multiplier for each student's score. Each receives the minimum of the group score and the multiplier times the group score. Thus, the most one can receive is the group score but a person who seems not to do their fair share can receive less. Also, I do still have quizzes and exams that are individual in nature.

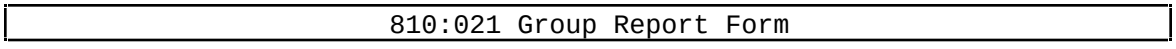
I am also considering blending the group proofs with the board work technique. Groups would bring their work to class and be assigned a problem for which they would transfer their proof to a transparency. In an effort to ensure whole-group learning, I would randomly choose a group member to present the proof. The class would receive feedback as to correct/incorrect and appropriate/inappropriate work and the group would a grade for their work **and** the student presenter's understanding of the proof.

I have considered using group work on (relatively short) programs also. Each student would write their own program, then the group would choose one program to revise and submit as the group program. The individual programs would be checked for correct output and the group program for quality. I haven't fully considered the ramifications, so do not suggest it as a technique. However, I do suggest it is appropriate to consider doing new things to improve the student learning.

Concluding Remarks

It is important to reiterate the motivation for these efforts—to enhance student learning. I sometimes catch myself getting a bit carried away with some technique or approach and have to remind myself to carefully consider whether I truly think a particular activity will enhance student learning.

My current bandwagon includes having students **do** the things I want them to learn rather than "know" something and for me to provide quality feedback on what they do. Quality feedback is both useful and timely. It seems that providing quality feedback requires that I modify my teaching practice as much as, or more than, my grading practice.



Use this report to keep track of your impressions of group member performance over the course of the semester. At the end of the semester, you will submit the report and it will be used in grading.

Identify and rate all the group members, including yourself. Enter a percent in the "% Work Done" column dividing 100% among the group members. For the other categories enter a number (0-4) indicating performance in the various areas. Use 0-1-2-3-4 to indicate:

never--------sometimes--often--usually--always

or unacceptable--marginal--okay---good-----excellent

GROUP MEMBER (list alphabetically)	% WORK DONE	ON TIME & PREPARED	QUALITY OF WORK	AID GROUP PROCESS
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____

Group: _____ Student: _____

Figure 1. A group report form.