# A NON-TRADITIONAL CLASS STRUCTURE FOR INTRODUCTORY PROGRAMMING, USING TABLET PC'S

**Chad Peiper** **Samuel N. Kamin**

*Computer Science Dept.*
*Univ. of Illinois at Urbana-Champaign*
*{peiper,kamin}@cs.uiuc.edu*

Networked Tablet PCs can dramatically change the classroom environment. In Spring, 2007, we taught an introductory programming course using a unique classroom structure made possible by Tablet PCs. Introductory programming is challenging to teach due to the very different levels of preparation with which students begin. A self-paced approach to instruction naturally suggests itself; however, it possesses fatal practical difficulties. With all students using Tablet PCs, their progress can be easily monitored, and this turned out to be the key to overcoming those difficulties. The result was a partially self-paced class, where the teacher could intelligently choose, at each moment, whether to lecture, help a particular student, go on to a new topic, or even put two students together to solve a problem. We describe what we did in the class, give the results of a student survey, and discuss further avenues for research.

## 1 PROBLEM STATEMENT AND CONTEXT

Introductory programming courses for Computer Science majors are notorious for the differential in preparedness of the students. While every class has students with differing levels of ability, these courses are nearly unique in the differing levels of *prior knowledge* of the subject matter. This makes these courses particularly difficult to teach – some students are lost, while others are bored.

Self-paced instruction is an obvious solution to this problem, but it relies upon having an entire lecture, with corresponding assessments, prepared ahead of time. More importantly, it can only work for certain students: most students still need to have a teacher explaining the material. Even if a teacher is present, she can only help one student at a time. The result is that there are still many students disengaged from the class.

The basic problem, then, is how to allow students to work at their own pace, while finding an effective role for the teacher – other than writing a programmed-instruction textbook.

## 2 SOLUTION EMPLOYED

In this study, the first author taught a special, 18-student section of our introductory programming course for CS majors, CS 125, in the Spring, 2007, semester. This section was taught in our experimental classroom (Figure 1), in which all students, and the instructor, used Tablet PCs. In this environment, the instructor could use a largely self-guided approach, yet easily and effectively monitor every student's progress. This ability was key to making the class work for the majority of students: he could determine what topic to lecture on at each moment, and could pair up the

Figure 1: Experimental classroom

students who were moving faster than average with those moving slower than average, thereby helping the slower ones while keeping the faster ones from getting bored.

The details of the class are discussed in the next section.

## 3 CS 125 CLASS STRUCTURE

We now describe the structure of the class. It must be noted that this structure does require additional work on the part of the teacher beyond what might normally be done (at least in a college course). Although the teacher lectures more or less as in an ordinary college course, the structure depends upon having a set of objectives and assessments against which students measure themselves during the class, and these must be prepared ahead of time.

For each class, the instructor prepared, and distributed to each student's computer, a sequence of objectives. Each objective was represented by a set of prepared questions which students encountered in this order:

- Pre-flight – one page, possibly several questions. Assesses whether students are prepared to go on to the assessment questions or must first hear a lecture on the objective. The student indicates whether he or she can answer those questions. An example of a pre-flight questionnaire is shown in Figure 2.
- Assessments – several self-study questions. The student attempts to answer each assessment, and is then shown the correct answer.
- Post-flight – a single question intended to sum up the objective and ensure the student has learned it; students are expected to be able to answer this question.

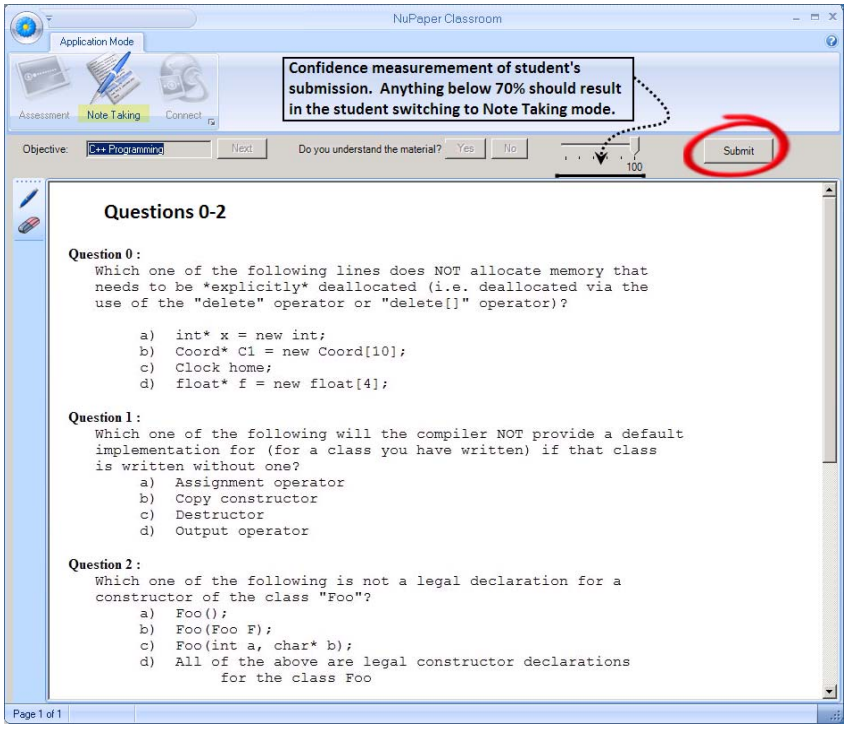
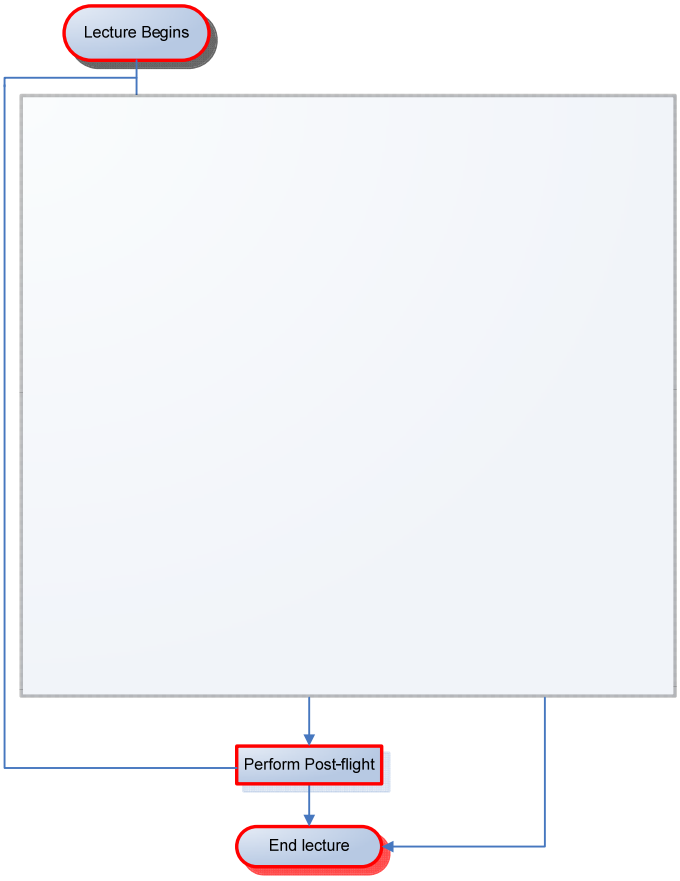


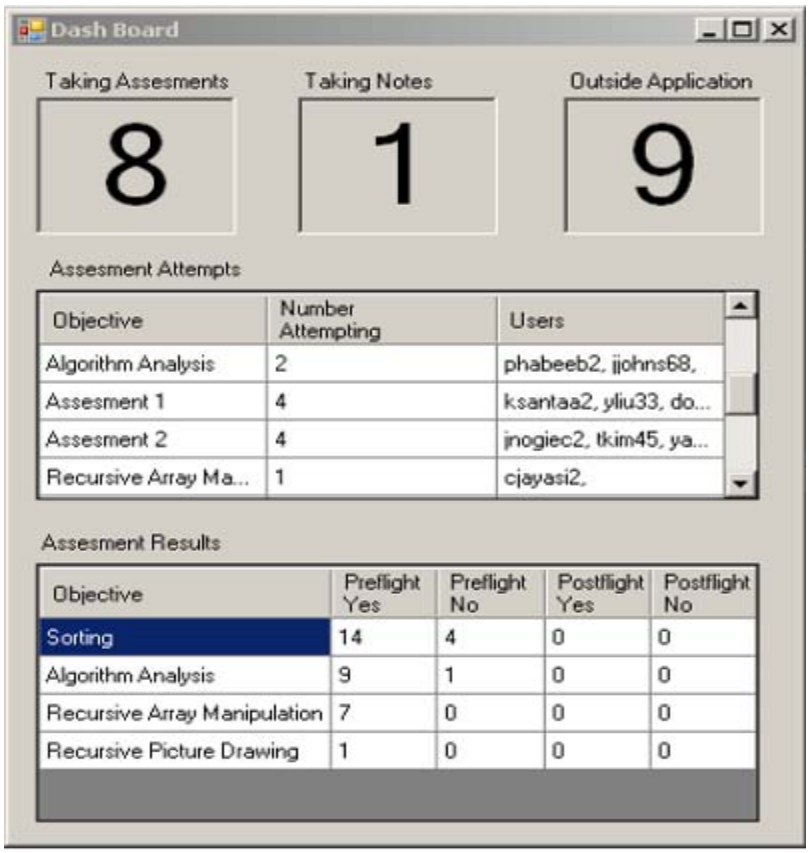

Figure 2: Sample pre-flight questionnaire

Note that each of the questions is self-graded, in the sense that the student sees the correct answer and compares it to his or her own answer. There is, in other words, no automatic grading (and the teacher obviously does not have time to grade these assessments in real time).

The flow of an entire class session is shown in the following figure. As shown, when a student indicates that she is unable to complete a pre-flight, she is placed in "note-taking" mode, which might also be called "listening-to-teacher-lecture" mode. If she does feel confident of being able to answer the pre-flight – meaning that she feels she already knows the material in this objective – she goes on to assessments. When she has answered those questions (to her own satisfaction), she can proceed to the pre-flight for the next objective.




Because students come to class differently prepared, and learn at a different pace, there will be students in various stages: some doing assessments (for various objectives), and some in "note-taking mode" (again, for various objectives). This is where the use of the Tablet PCs is essential, because it allows the instructor to know where each student is, and make decisions accordingly. Specifically, the instructor sees the state of the class on a "dashboard" like this one:




Armed with this information, the instructor can make a choice among a variety of alternatives. Suppose the entire class has gone past objective *n*:

- If the majority of students are in note-taking mode for objective *n+1* – that is, they have reported that they cannot answer the pre-flight for this objective – then the instructor will lecture on that topic. (Objectives are designed to be taught in 10-15 minutes.)
- If the majority of students have passed beyond objective *n+1*, but a small number of students are still in the assessments phase, the instructor may repeat the lecture for them (or provide individual assistance, if there is just one such student). He may also force them to objective *n+2*. (They can return to objective *n+1* after class.)
- If a student has gone past objective *n+2*, he may pair him with a student working on an earlier objective. This may involve physically moving the students (and their Tablets) to be next to each other.

The overall effect is that the instructor can lecture to the majority of the class, on just the topic that they need to learn, offer individual assistance to students when needed, and keep the more advanced students from disengaging. He may not be able to address every student's needs optimally, but the dashboard allows him to make intelligent decisions about where to address his efforts.

## 4 EVALUATION

At the end of the spring semester, we administered a survey to assess the students' attitudes toward the technology. (Because the instructor was also a researcher, there was little point in asking him to fill out a survey.)

Several of the questions asked about specifics of the class, or asked speculative questions about possible future enhancements of the technology; we omit those for lack of space. The ones that directly asked students to compare the class with lo-tech classes were these:

> **1**. The classroom lecture was more engaging or interesting to me as result of the use of technology in class and the availability of notes afterwards.
> **2**. Because captured lecture notes are available after class, I was able to better pay attention to the lecture.
> **10**. All things being otherwise equal, I would prefer to take a class that uses Tablet PC technology over the same class that does not.
> **11**. Tablet PC technology will encourage students to skip lectures.
> **12**. Tablet PC technology made me less worried about missing class.

This table gives the results from this survey. Clearly, the students had a positive feeling about the technology. (One student was not able to complete the survey.)

| Question | Strongly agree | Agree | Neutral | Disagree | Strongly disagree | No answer |
|---|---|---|---|---|---|---|
| 1 | 2 | 10 | 2 | 2 | 1 | 0 |
| 2 | 2 | 2 | 10 | 2 | 1 | 0 |
| 10 | 3 | 8 | 2 | 0 | 2 | 2 |
| 11 | 1 | 2 | 8 | 4 | 0 | 2 |
| 12 | 0 | 4 | 6 | 3 | 2 | 2 |

## 5 FUTURE WORK

Since this work was done, we have been studying "teacher's dashboards" in more detail (to be reported in Peiper's forthcoming PhD thesis). In studies this year, we have used dashboards in high school classes. The important point about these newer studies is that we have attempted to provide a benefit to the teacher – improved transparency of the classroom – without requiring a change in teaching methods or development of new materials. That is, while the current study points to new ways of structuring the classroom experience, we believe that a dashboard can be a useful adjunct to *any* classroom. Further, we believe this can be a particularly fruitful field for research. The broad issue is how to present the activities of an entire class to the teacher so that she can obtain actionable intelligence quickly (even in large classrooms). This involves deep questions of visual presentation, user-interface design, and computer analysis of handwriting data.

## 6 ACKNOWLEDGEMENTS

The authors wish to thank Boris Capitanu, who provided substantial help with the programming for this experiment.