

Test Taking Software

For most of the year, being a professor is better than working for a living. The exception is grading exams. Part of the problem is that that is when I find out how bad a job I did, how many students did not learn what I thought I had taught them. Part is the time and trouble of evaluating their answers and of trying to read their handwriting. The last is the part that can be solved and to some degree has been.

More than thirty years ago, when I was teaching at Tulane Business School, I came up with an idea for a way to make it easier for students to take exams and professors to grade them. It is easier to write with a word processor than a pen and much easier to read what has been written. Have a roomful of terminals or desktop computers — this was long before students had laptops — running a simple word processor. The exam questions are on the screen, the students type in their answers. I called it Electric Bluebook.

To make things easier, the student would see a page listing questions and letting him mark each as not yet done, completed, given up on, or to be looked at again if there is time. When the students were finished, all of the exams would be transferred to the professor's computer for grading. It is easier to comment and do a fair job of grading if you first grade all answers to question one, then to question two, but with paper exams doing that is a pain. The computer eliminates the problem of shuffling bluebooks.

A professor may get more tolerant of poor answers after seeing a lot of them, giving an advantage to the students graded last, or more intolerant, giving a disadvantage. A professor may form an opinion of a student based on his answer to one question and let that distort the grade on the next.¹ To solve both problems, give him the option of having the software feed him the answers to each question in random order. Electric Bluebook never got to completion, as has happened with a number of my projects. But many years later other people did create such software, this time running on student laptops.

My objective was to make the process as easy as possible for both students and professors. Examsoft, the software that the university I last taught at used, had a different priority. It was designed to solve one and only one problem: cheating. The programmers faced, and believed that they solved, the difficult problem of how to let a student use their software on his own laptop while locking him out of everything else on the machine, which might include class notes, answers to old exams, even a downloaded textbook. Also locking him out of the internet. I liked my design better but understand the depressing reasons for theirs.

That started me thinking of alternative approaches to their objective. A school could buy a bunch of cheap laptops dedicated to the sole purpose of taking exams on, wiped clean of everything else after each use. That would get them back to something like the setting I originally assumed, where Electric Blue Book would have been running on hardware belonging to and controlled by Tulane, not the student.

¹ Daniel Kahneman in his (very good) *Thinking Fast and Slow*, describes an experiment he did on his own grading. He found that doing it question by question instead of exam by exam resulted in the performance of students coming out less consistent, made it less likely that the student who got a good grade on question one also got a good grade on question two. He concluded that he had a strong bias towards imposing consistency on his grading, even when it was not there in the students' performance.

Twelve years after I wrote my blog posts on the subject and thirty-some after I came up with the idea for Electric Blue Book, I finally got around to searching the web to see what other people had done.² There is a fair amount of test taking software out there, much of it running on the Internet. Many of the programs claim to have protection against cheating, in some cases automated, some with remote proctoring. None, so far as I could judge by their webbed description, made any serious effort to do what I wanted, to make the whole process easier for student and professor.

When I blogged about some of this, I got comments from people who knew more about computers and the problems of computer security than I did. Some argued persuasively that what Examssoft and its competitors claimed to do was not doable, at least not if sufficient ingenuity went into defeating it. The software can, Examssoft does, take over the computer that it is running on in order to prevent the user from accessing material on his hard drive or online. To evade that, create a virtual computer on your physical computer, load Examssoft on that, and switch between the padlocked virtual computer and the unlocked physical computer at will.

I do not know what precautions the designers of the existing software have taken to defend against that and other attacks. My guess is that the protection provided by the existing software would work against the average test taker, which may as a practical matter be sufficient, but not against a sufficiently sophisticated user. If so, the following alternatives are available:

1. Use something like Electric Blue Book and trust your students.
2. Use something like Electric Blue Book and watch your students. As one commenter put it:

How about a hardware solution rather than a software solution: place the teacher's desk behind the students, rather than in front of them.

3. Use something like Examssoft, perhaps an improved version with some of my features built in, on the assumption that not many of your students will be able to figure out how to hack it.³
4. Write exams designed to be taken open book and open computer.

As another commenter put it:

I consider "but the student can Google the answer" to be a feature, not a bug: make up questions that require them to think, instead of spitting back memorized trivia.

² And discovered that there was a program out there, widely used, called "Electronic Blue Book." The earliest reference I found to it was from 2004, about twenty years after my project. I like my title better, "electric blue" being a color.

³ The history of copy protected software in the early years suggests that eventually someone will write and circulate an Examssoft cracker, written by a sophisticated user to be shared with unsophisticated users. It might end up as an arms race between the programmers of Examssoft and the programmers of the cracking program.