

Using the Internet to Facilitate Examination of Student Proficiency

J. A. Murden, K. P. Brannan
The Citadel

Abstract

In the fall of 1997 the authors began to use the Internet both to deliver some examinations to students and for students to return solutions to the examinations back to the professors. After an initial period of apprehension on the part of some students, the students in the classes responded favorably to web-based tests. Students overwhelmingly stated that the web-based tests were no more difficult and did not affect the grades compared to the number of other types of tests used in the course. Initially, it took almost three hours to plan and create a web-based test. With experience, the required time was quickly reduced to the same one-hour period that had been required for the traditional tests. The time saved grading the web-based tests and assessing student performance was substantial and resulted in a net decrease of faculty time for the case presented in this paper. The web-authoring software, Microsoft FrontPage, used to create these tests was moderately priced and compatible with the software routinely used by the authors. The software and the computing infrastructure that supported this effort are described.

Introduction

What engineering professor hasn't puzzled far too long over a partially legible, weakly organized test in an effort to fairly evaluate student proficiency, while concerned about timely feedback? Frequently faced with a stack of such papers to grade, it would be natural for a professor to consider whether more efficient means are available for assessing tests. A logical alternative to traditional testing that has the potential for promoting efficiency is digitally delivered tests. Computer-based testing offers a professor a number of advantages over traditional tests, including rapid grading and feedback, the potential for adding multimedia elements to tests, and an effective means to assess student work and courses. Further, the Internet offers a reliable way to deliver and receive tests on campus, to homes of non-resident students, or to any location at any time in a distance learning environment.

Computer-based testing has been explored for a number of years, but until recently, the resources were not generally available for an individual professor to develop creative computer-based tests on a regular basis. As a graduate student in 1985, the lead author wrote a suite of programs that allowed elementary school teachers to create and administer spelling or vocabulary tests by using the audio capabilities of the (then revolutionary) Amiga computer. *Quizzer*¹ (a network-based, non-internet tool) and *InternetQuiz*² are two special purpose software programs that have been developed in recent years to facilitate the administration of computer-based tests. Internet-based tests can also be conveniently developed and managed using web-authoring software such as FrontPage.

The purpose of this paper is to discuss the experiences and lessons learned using computer-based tests in a first course in computer programming. Of particular interest is the design and implementation of tests that are downloaded via a web browser during class, completed by the student and submitted before the end of the class period. The time required by the authors to plan and create suitable tests for this format is compared to more traditional paper tests. The techniques and software tools that have been used to create these tests are also discussed. Further, the computing infrastructure needed to support this effort is detailed.

Background

During the 1994 Fall Semester, in an effort to improve student proficiency in programming and promote enthusiasm in computer use, the authors began voluntarily team-teaching in the Civil and Environmental Engineering Department's computer applications course. In addition, the authors began to gradually experiment with techniques for delivering course material, with testing methods and the number of tests given during the course. In Fall 1996, changing the computer language from C++ to the scripting language in Mathcad made a significant difference in computer use throughout the department and in overall student enthusiasm³. As the course evolved, a wide variety of teaching techniques was incorporated, including traditional lectures, cooperative learning, hands-on activities in the computer laboratory with "electronic workbooks" developed by the authors, lectures and demonstrations in a multimedia classroom, and individual work sessions. Additional variety was achieved by rotating lecture responsibilities between the two professors or by creating a classroom setting with both professors interacting. More individual attention was possible with two professors to answer questions during work sessions.

When the authors began team-teaching the course, three tests and a final examination were given during the semester. Later, the testing frequency was increased to a weekly basis. Student response to frequent testing was remarkably positive. With more frequent testing, the authors began to consider ways to facilitate delivery, submittal, and grading of tests. Because rapid assessment of student understanding was especially important, the possibility of using the Internet as a means of delivery and submittal was explored. In the fall of 1997, the authors decided that they had gained adequate network experience and that sufficient resources were available within the department to support this venture. Therefore, a number of goals were established to guide the efforts:

- Determine if special purpose software must be obtained or if the department's existing web authoring and management software would be adequate for generating, delivering, and receiving tests
- Determine if computer-based tests administered over the Internet would reduce the faculty time and effort spent on testing and assessing student performance in a course where tests are given once a week
- Assess whether computer-based tests have an impact on student performance
- Develop a system that could be applied to both resident and evening (non-resident) student populations at The Citadel

Test Styles

To provide a basis for comparison, several test styles were used in the computer applications course. One was a traditional test given on paper. Questions on this type of test ranged from providing explanations to writing programs. A second test type involved taking a “hands-on” test in Mathcad in which a student performed operations and developed solutions on a Mathcad worksheet. The test containing the problem description was delivered to students in two ways: (a) on paper and (b) downloaded from the department’s server as a Mathcad file via a web browser. Completed worksheets were printed on a network printer in the laboratory and submitted. The third test style involved downloading an html form into a web browser. The test was conducted in the browser environment and the solution was submitted back to the department’s server. While this type of test was not deemed advantageous for writing extensive programs or for performing Mathcad operations, it turned out to be reasonably versatile for creating tests for much of the material covered in the course.

Testing packages have been developed that have the capability to evaluate the student responses, then immediately provide the grade and feedback to the test taker^{1,2}. However the authors have not yet attempted to implement those features in the web-based tests.

Test Design, Construction and Implementation

Creating a web-based test is very similar to creating any other web page with web authoring software. The overall process is shown in Table 1. One of the most worrisome concerns the authors faced at the start of this project was how to translate the types of test questions that are appropriate for the computer course into a format suitable for the web-based tests. The most easily presented types of questions and methods of implementing them are shown in Table 2. At first glance, this list may appear limiting. In fact, this short list is quite flexible and ultimately, the resulting computer-based tests were considered to be equivalent to the other test styles that had been used previously. The authors have used the question types in Table 2 to ask students to predict the output produced by a program, to "write" short segments of code, or to select which graph would be generated by a sequence of statements. Examples of some test questions using three of the types of questions are shown in Figure 1. In general, if a question can be presented in a manner that allows an adequate response to be entered using standard keyboard characters or by selecting options with a mouse, then the question may be conveniently utilized in a web-based test.

| |
|--|
| Design the test |
| Create the html form for the test in FrontPage |
| Create the test questions |
| Arrange the elements of the test in the form |
| Save the test form |
| Administer the test |
| Open the test results file and extract the solutions |
| Return the graded test solutions to the students |

Table 1. Web-testing using FrontPage

| Question Type | Implemented by: |
|---------------------------------|--------------------------------|
| True-False, multiple choice | radio buttons, drop-down lists |
| Fill-in-the-blank, short answer | non-scrolling text box |
| Discussion or Essay | scrolling text box |

Table 2. Easily utilized types of test questions.

FrontPage⁴ is the web document creation and site management software used by the authors. It allows the html forms required for the web-based test to be generated in a point-and-click environment. This allows the test creator to focus on the document rather than the production of the file. The authors prepared their first web-based test in two to three hours. The second test took about an hour and a half and subsequent tests were completed in less than one hour (about the same time as the traditional tests).

A convenient starting point for creating a test is to use the FrontPage Editor “survey form” template to create the html document. The template contains fields for respondent supplied information and a submit button. Appropriate configuration of the “submit” button will cause a test solution to be appended to the designated solutions file on the server. FrontPage, through a series of "extensions" loaded onto the web server, creates the files on the server to receive the test results from the students and creates the required links in the test document to send the data in the form specified by the author of the test.

Most of the preparatory work for the tests is the same for traditional or web-based tests. Therefore the same word processor, computer-based drawing and mathematics packages are used in both cases. If hand-drawn figures or existing illustrations are preferred, they can be scanned and inserted into a web-based test just as easily as a traditional document. Inserting the text, graphics, equations, radio buttons, etc., in “Tables” (conveniently done within the FrontPage Editor) facilitates the arrangement and presentation of the items on the test.

In the initial phases of this project, security issues were raised by some of the authors' colleagues. For this project, test security was promoted in several ways. Links to tests were not created from web pages; the URL was provided at test time. File names were randomly selected for the test documents. Before students began leaving the classroom, the web-test file was renamed. Together, these precautions make unauthorized access unlikely. Occasionally, different versions of the test were created. Because of the arrangement of computers in the laboratory, random ordering of test questions on individual tests was not deemed necessary. The authors considered these measures to be more than sufficient for the course. If more security

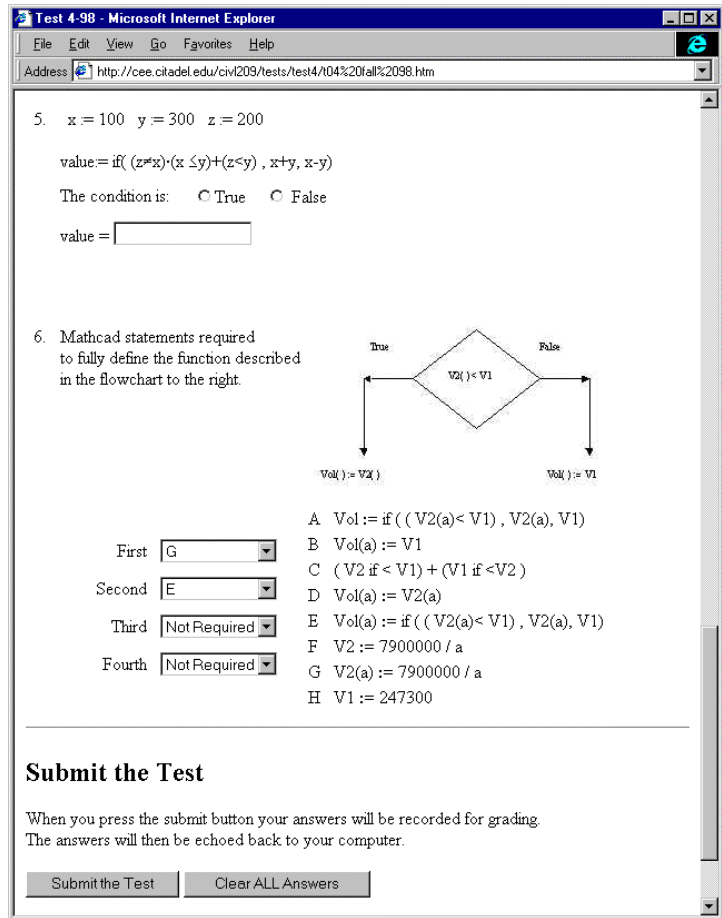


Figure 1. Typical Web-based Test Questions

precautions are desired, perhaps for distance learning applications, several additional requirements may be easily instituted. The FrontPage server extensions provide the capability to schedule the times that a test file is available from the web server. Further, if so desired, the test taker can be required to supply a password before accessing the test file.

Although not strictly a security measure, the server extensions can also require that specified information fields (such as: name, student ID number, section number) are completed before a web-based test solution can be submitted to the server. As a result, no tests are received without names or other required identifying information. This student-supplied data can be used with additional information that can be gathered by the server: the time of the submission, the IP address of the computer used, a username, etc. to validate the submittal.

At the professor's discretion, a confirmation copy of the student's answers may be transmitted back to the student's computer. The student may print this form and compare it to an answer key for instant feedback. Following the test, a professor accesses the server file containing the test solutions to begin the grading process. From the many formats FrontPage allows for solution files, the authors used "formatted text" most often. Such a file can be opened in a word processor and (a) graded directly in the word processor and e-mailed back to the student or (b) printed, graded by hand, and returned to the students at the next class. Alternatively, although the authors have not yet utilized this option, the solutions file can be used as the input to a computerized grading program to evaluate the objective portions of tests.

Computing Infrastructure

At The Citadel, the Information Technology Services (ITS) department develops and maintains the campus-wide computing infrastructure. ITS provides standard college-wide solutions to meet the general needs of students, faculty and staff computing needs while giving flexibility to the faculty who develop and implement custom solutions when those are more appropriate. The campus-wide network connects faculty offices, computer laboratories and on-campus student housing to the ITS computer systems and the Internet. Each computer in the CEE faculty offices and in the CEE computer laboratory has a 10 MB/sec Ethernet connection to the 100 MB/sec campus-wide backbone. The CEE server is on the same segment as the computer laboratory, which eliminates most data transfer delays.

The CEE Department's main teaching computer laboratory is equipped with twenty-three Gateway 333 MHz Pentium II computers with: 64 MB RAM, 2 GB hard drives, CDROM Drives, sound systems with headphones. These computers were installed at the beginning of the Fall 1998 semester replacing the same number of 90 MHz Pentium machines containing 16 MB RAM; 1 GB hard drives. (The 90 MHz machines did not limit any efforts in this project.) Netscape and Microsoft web browsers are installed on the laboratory computers.

The CEE department had previously licensed Microsoft NT Server 4.0 and Microsoft FrontPage 97 to host the department's web sites. FrontPage 97 is used to administer the web server and the individual web sites. As previously discussed, FrontPage is also the primary html document authoring tool. The server is administered by the lead author and runs on his desktop computer, a 166 MHz Pentium-based computer with 64 MB RAM and 3 GB of hard disk space. This computing system usually meets the server and faculty workstation demands without significant performance penalties.

Student Reaction

Every effort was made to ensure the content of the web-based tests was the same as the traditional tests. There was some question in the beginning of this project about student reactions to the use of drop-down lists, radio buttons or check boxes in the presentation of the test questions.

The authors reasoned that the students in a computer-programming course would probably accept any computer-based task with the same

level of tolerance (or intolerance) and decided to ask for student opinions after the experiment was well underway. Two groups of students were asked the same short set of questions. One group of 27 students was asked to complete the survey after taking a single web-based test. A second group of 20 students was surveyed after taking five web-based tests. The survey questions and the number of each of the responses are presented in Table 3. Even those students who had some apprehension about taking the web tests did not report thinking the level of difficulty was greater or that their scores were lower.

Student comments from the survey were overwhelmingly positive. Some students "enjoyed the change" and said "felt it made me think more" or "seemed a cleaner test". One student replied, "I personally enjoyed the web-based tests. I found that they were easier to follow. The standard tests were sufficient. But the web-based ones were more expedient and convenient for all involved."

The negative comments were almost exclusively focused on the apprehension factor. About two thirds of those negative comments were associated with the mechanics of completing or submitting the web-based tests. As a result, the authors will use a practice test in the future before the first web-based test is given in a course.

Concluding Comments

- ❑ Readily available, moderately priced software such as FrontPage may be used in conjunction with a faculty member's preferred text and graphics processing programs to design creative and effective web-based tests that are comparable to other test styles in degree of difficulty.
- ❑ The authors required nearly triple the preparation time of a traditional test to generate the first web-based test with FrontPage. However, the skills and experience needed to create and

| Student Survey questions | Number of responses from those | |
|--|---|---|
| | taking five tests | a single test |
| Did you recall taking web-tests? | 20 - yes 0 - no | 26 - yes 1 - no |
| Were you apprehensive about taking the web-based tests? | 3 - yes 17 - no | 11 - yes 15 - no |
| By taking the web-based tests, do you think your grade was? | 4 - raised 13 - not changed 2 - lower | 5 - raised 17 - not changed 4 - lower |
| The level of difficulty of the web-based tests relative to other tests in the course was | 0 - higher 16 - no change 4 - lower | 1 - higher 20 - no change 1 - lower 4 - did not recall |

Table 3. Student survey questions and responses.

administer web-based tests can be acquired rapidly. The third web-based test was created as quickly as a traditional paper test.

- All solutions recorded during web-based tests are legible and arranged as desired by the professor, which simplifies and speeds the task of scoring the tests. The quicker "turn-around" of the tests allows for more rapid feedback to students and professors.
- Test responses submitted in computer accessible form facilitate the documentation of both individual student performance and of courses. The authors see documentation as one of the significant tasks supporting the assessment expected by institutions, state agencies and ABET EC 2000⁵. Because the information is in electronic form, it may readily analyzed by a variety of methods to help promote continuous improvement of teaching and learning.
- Professors with moderate computing skills and administrative privileges on a web server can easily administer web-based testing to resident student populations. The authors expect web-based tests to be just as successful with non-resident student populations.
- The overall student response to the web-based tests given in this project was positive. The initial apprehension reported by some students was associated primarily with the mechanics of completing or submitting the test and can be reduced by giving a practice examination.

References

1. Jung, Il-Hong, Ku, Hosoon, and Evans, D. L., "A Network-Based Multimedia Computerized Testing Tool," Session 3530, *Proceedings of the American Society for Engineering Education Annual Conference*, Milwaukee, Wisconsin, 1997.
2. Crepeau, R. G., "Student Assessment with InternetQuiz," *Proceedings of the American Society for Engineering Education Annual Conference*, Seattle, Washington, 1998.
3. Brannan, K. P. and Murden, J. A., "From C++ to Mathcad: Teaching an Introductory Programming Course with a Non-Traditional Programming Language," Session 1253, *Proceedings of the American Society for Engineering Education Annual Conference*, Seattle, Washington, 1998.
4. FrontPage 97, Microsoft Corporation, 1997.
5. Engineering Accreditation Commission of The Accreditation Board for Engineering and Technology, *Engineering Criteria 2000*, Third Edition, Baltimore, MD, 1997.

JOHN ALDEN MURDEN is an associate professor of Civil and Environmental Engineering at The Citadel. He earned the Ph.D. from Clemson University while studying wind effects on structures. Dr. Murden's technical interests currently include the effective application of computers in engineering education, system modeling, and experimental methods in structural mechanics. He is actively involved in the Southeastern Section of ASEE.

KENNETH P. BRANNAN is an associate professor of Civil and Environmental Engineering at The Citadel. He is currently President of the Southeastern Section of ASEE. He earned B.C.E and M.S. degrees from Auburn University and the Ph.D. from Virginia Tech. A registered professional engineer, Dr. Brannan has interests in freshman engineering education, computers in engineering education, water supply and wastewater treatment systems.