Practice Makes Perfect: A simple Javascript Routine for Student Practice Tests that Anyone can Implement

Gary B. Randolph, Dewey A. Swanson, Dennis O. Owen & Jeffrey A. Griffin Purdue University School of Technology – Anderson/ Purdue University School of Technology – Columbus/ Purdue University School of Technology – South Bend

Abstract

Beginning in the spring 1999 semester and continuing, the authors have used a simple Javascript routine to create web-based practice tests for students in various classes. Feedback from students has been very positive. Students indicate that the question-feedback cycle helps them gauge their readiness and feel more confident for the real exam.

This paper will discuss how the program works, how educators can enter customized test questions for the program using a simple text editor, and provide a web link for educators to use to download the program and customize it for their own classes.

I. Introduction

Computer software has been put to many different uses in education, including tutorials¹, instruction delivery², simulations³, and practice tests⁴. With the growth of the Internet, on-line practice tests have the advantage of allowing students to access the practice test at any time, allowing students to progress at their own pace, providing instant feedback, and allowing information to be constantly updated⁴. Students using online exams report that they appreciate the immediate feedback on their performance as well as the "anytime / anyplace" capability⁵.

Javascript is a simple non-compiled programming language for creating web pages.⁶ Javascript was used to create a practice test web page for students to use in preparing for exams. The fact that Javascript is non-compiled means anyone can easily modify practice tests for their own classes using a simple text editor program, as the authors have done. Being non-compiled also means that the questions and answers are not secured; any user who knows how to view the source can do so. But for the purpose of a non-graded practice test, that is of little concern. Students run the practice test to learn and gauge their knowledge, not to get a grade.

II. Taking the Test

When students start the practice test program they will first see a welcome screen telling them what to expect. When they click on the Start button they will then begin seeing questions as shown in Figure 1. The program is set up to select a random group of questions for presentation

from a larger universe of possible questions. This encourages students to take the practice test several times and to thus drill themselves on the information.



Figure 1. Sample Question

Students can click on any of the three answers. Their selection will lead them either to a "Wrong Answer" screen as shown in Figure 2 or a "Correct Answer" screen as shown in Figure 3. In either case they will be told what the right answer is and why.





Figure 3. "Right Answer" Screen

When students finish the practice test, they will be given a summary of their performance. They will also be given an opportunity to take the practice test again with a different random selection of questions.

III. Customizing the Test

The Javascript program resides in a standard HTML web document. The document can be opened and edited in any text editor, including Windows WordPad and Notepad. Figure 4 shows the first part of the document, which needs to be edited for customization. It is recommended that a backup copy of the document be made prior to editing, in case anything goes wrong. Javascript is more than a little picky. A forgotten quotation mark can cause the entire quiz to fail. The following edits must be made.

On the line that says "var numquestions =" change the number to the number of questions you will actually be including in the quiz.

On the line that says "var num2ask =" change the number to the number of questions you want asked in each session. For instance, you might have 30 questions entered, but have the quiz set up to ask a random sample of 10 of those questions each time as student runs the quiz. This will encourage students to run the quiz multiple times, providing repetition with many of the questions.

On the line that says "var numMsg =" enter the number of wrong messages you want the program to choose from when responding to a wrong answer. The actual messages are listed below in the //Wrong Messages section. You can delete any of those messages you don't like and add others. But make sure that (1) the remaining messages are numbered sequentially from 1, and (2) the total number of messages matches the numMsg entry in the top section.

Figure 4. Javascript Program Heading

```
<html>
<head>
<title>Practice Study Quiz</title>
<script language="JavaScript"><!--
//This JavaScript code is copyrighted (c) 1998 by Gary Randolph - gbrandolph@purdue.anderson.edu
var emptyFrame = '<html></html>';
var rand = new RandomNumberGenerator()
var numquestions = 20
                                           //number of questions in data
                                           //number of questions to ask
var num2ask = 5
var currquestion = 0
                                          //current question #
                                          //number answered correctly
var numright = 0
var numasked = 0
                                          //number of questions asked
var numMsg = 7
                                               //number of wrong messages
var exitpage = "http://www.purdue.anderson.edu/cpt/courses/cpt145/cpt145_info.htm"
                                                                                       //page to
exit to
        //Base Arrays
       Question = new createArray(numquestions,'')
       AnswerA = new createArray(numquestions,'')
       AnswerB = new createArray(numquestions,'')
       AnswerC = new createArray(numquestions,'')
       CorrectAnswer = new createArray(numquestions,'')
       Explanation = new createArray(numquestions,'')
       QPic = new createArray(numquestions,'')
       APic = new createArray(numquestions, '')
       Used = new createArray(numquestions, '')
       Used[0] = 1
       WrongMsg = new createArray(numMsg,'')
       //Wrong Messages
       WrongMsg[1] = "Well, you were close. But the right answer is: "
       WrongMsg[2] = "Not guite. The right answer is: "
       WrongMsg[3] = "That may seem like a good answer to you, but the real answer is: "
       WrongMsg[4] = "I hate to not be affirming, but the correct answer is: "
       WrongMsg[5] = "That is an interesting way to look at it. But a better answer is: "
       WrongMsg[6] = "Not exactly. The real answer is: "
       WrongMsg[7] = "That's a good choice -- unless you want to be right. The real answer is: "
       // Ouestion 1 info
       Question[1]="ESS is the acronym for..."
       AnswerA[1] = "Electronic Support System"
       AnswerB[1] = "Enterprise Support System"
       AnswerC[1] = "Executive Support System"
       CorrectAnswer[1] = "C"
       Explanation[1]="An Executive Support System (ESS) is designed to be user-friendly systems
that pull together all the information executives need."
       QPic[1] =" "
       APic[1] =" "
```

On the line that says "var exitpage =" enter the web address of the page that you want the quiz to exit to after students complete the quiz. This should be the class home page, if any.

The //Question X Info sections provide the information for asking and answering the quiz questions. There is one section for each question that can be asked. The quiz as initially downloaded has 20 questions set up. If you use fewer than 20, you can delete the others. If you want to add more questions, copy and paste one of the //Question X Info sections, renumbering the subscripts in brackets [] appropriately. The various variables of the //Question X Info sections are:

• For Question[x] enter what you want the students to see as the question.

- For AnswerA[x], AnswerB[x], AnswerC[x] enter the three possible answers. Currently the quiz is limited to multiple-choice questions with three possible answers. This keeps the quiz simple for practice situations.
- For CorrectAnswer[x] enter either "A" or "B" or "C." Notice that the answer is enclosed in quotation marks.
- For Explanation[x] you can put in an explanation of why the correct answer is correct; this will be shown as feedback.
- Qpic[x] and Apic[x] are optional. They are path and file names for jpgs or gifs that can be shown. Qpic[x] is shown with the question, Apic[x] with the answer. If you use them make sure they are installed on the web server in the correct directory. A good practice is to install the pictures in the same directory as the quiz document. Then you don't have to include a path name with the file name.

Once the above modifications have been made, the file simply has to be saved, tested, and copied to a web server. Create a link to the practice test program from the class web page.

IV. Obtaining the Test Program

Instructors can run a sample version of the practice test program using a web browser and the address: <u>http://www.purdue.anderson.edu/download/jquiz/quiz.htm</u>. To download a copy of the quiz, go to <u>http://www.purdue.Anderson.edu/download</u> using a web browser and click on "Javascript web-based practice test." You will need the Winzip program to unpack the files. A shareware version of Winzip can be obtained from a link on that same page. There are three files in the zip file: the HTML file containing the Javascript routine, another HTML file called Empty that is needed to set up the frames, and an exitbutton.jpg graphic that is used as an Exit button in the program. All files should be placed in the same directory on the web server.

V. Results

Informal feedback from all classes in which the quiz has been used indicates that it is popular with students. One of the authors surveyed students after using the quiz to prepare for an exam. Thirteen of the fourteen students in the class used the quiz, with seven of them running it ten or more times. All thirteen reported the quiz to be easy to use. Twelve of the thirteen said it was useful in reviewing for the exam.

VI. Ideas for Improvement

As easy as the Javascript program is to modify, it could still be easier. One idea for version 2 of the program is to allow instructors to enter the questions and answers into a companion Microsoft Access database with a function built in the database to automatically create the Javascript program. That would make the design process easier and more foolproof. Some students running the quiz found that the random assignment of questions repeated many of the same questions each time they ran the quiz and would like that repetition eliminated. Some students requested the inclusion of True/False and fill-in-the-blank questions.

Bibliography

 Barrett, S. F., Pack, D. J., York, G. W., Neal, P. J., Fogg, R. D., Doskocz, E., Stefanov, S. A., Neal, P. C., Wright, C. H., & Klayton, A. R., "Student-Centered Educational Tools for the Digital Systems Curriculum," <u>Computers in Education Journal</u>, pg. 6-11, Vol. IX No. 1.
 Boulet, M., & Boudrealt, S., "Using Technology to Deliver Distance Education in Computer Science," <u>Engineering Education</u>, pg. 433-436, Vol. 87, No. 4.
 Brooking, C. J., & Smith, D. A., "Simulation and Animation of Kinematic and Dynamic Machinery Systems With Matlab," <u>Computers in Education Journal</u>, pg. 2-3, Vol. IX No. 2.
 Ng, A, & Gramoll, K., "Online Review and Practice Tests for the Fundamentals of Engineering Exam," In <u>1999 ASEE Annual Conference Proceedings</u> [CD-ROM], ppg. 13.
 Crepeau, Raymond G. "Student Assessment with Internetquiz," In <u>1998 ASEE Annual Conference Proceedings</u> [CD-ROM]. Seattle, WA: American Society for Engineering Education, ppg. 3.

6. Reynolds, M. C., Special Edition Using JavaScript, Indiapolis: Que (1996).

GARY B. RANDOLPH

Gary Randolph is an Assistant Professor in the Department of Computer Technology at Purdue University School of Technology in Anderson. He is a member of ASEE and the Association for Information Systems. He earned a BS at Taylor University and an MA at Ball State University. He has eighteen years experience in information systems development. In addition to teaching he maintains an active consulting practice, specializing in database and web site development.

DENNIS O. OWEN

Dennis Owen is a tenured assistant professor of Computer Technology at Purdue University, Anderson, Indiana campus. He earned his B.S. and M.A. at Ball State University. He spent ten years in the automotive industry prior to his appointment to the Purdue faculty. He is a member of ASEE and a charter member of the Association for Information Systems. His primary teaching interests are operating systems and programming languages.

DEWEY A. SWANSON

Dewey Swanson is an Associate Professor of Computer Technology at Purdue University School of Technology, Columbus, IN campus. He received his B.S. degree in Secondary Education from Indiana University in 1978. He received his M.S. degree in Mathematics from Indiana State University in 1980. He worked 14 years in industry in various computer-related functions.

JEFFREY A. GRIFFIN

Jeff Griffin is an assistant professor of Computer Technology for the Purdue University School of Technology site at South Bend, Indiana. He came to Purdue University in 1997 after a fifteenyear professional career in the U.S. Navy and with Electronic Data Systems, Inc.