



ISSN: 1545-679X

Information Systems Education Journal

Volume 4, Number 89

<http://isedj.org/4/89/>

October 4, 2006

In this issue:

An Alternative Testing Strategy for Advanced Programming Courses

Wendy Ceccucci

Quinnipiac University
Hamden, CT 06518 USA

Abstract: The traditional methodologies for evaluating student learning have been either a hand-written exam, usually in the form of multiple choice questions, or a take home program. These methods both have limitations which make it difficult to accurately measure a student's ability to apply the material learned to design and develop a program. An alternative testing strategy would be to let each student develop a take-home program exam for another student in the class. This evaluation strategy would allow students to uniquely apply the material learned. While developing their own programs, students would gain a better perspective of the limitations of their current knowledge.

Keywords: assessment, testing, exams, programming assessment, advanced programming

Recommended Citation: Ceccucci (2006). An Alternative Testing Strategy for Advanced Programming Courses. *Information Systems Education Journal*, 4 (89). <http://isedj.org/4/89/>. ISSN: 1545-679X. (Also appears in *The Proceedings of ISECON 2005*: §2343. ISSN: 1542-7382.)

This issue is on the Internet at <http://isedj.org/4/89/>

The **Information Systems Education Journal** (ISEDJ) is a peer-reviewed academic journal published by the Education Special Interest Group (EDSIG) of the Association of Information Technology Professionals (AITP, Chicago, Illinois). • ISSN: 1545-679X. • First issue: 8 Sep 2003. • Title: Information Systems Education Journal. Variants: IS Education Journal; ISEDJ. • Physical format: online. • Publishing frequency: irregular; as each article is approved, it is published immediately and constitutes a complete separate issue of the current volume. • Single issue price: free. • Subscription address: subscribe@isedj.org. • Subscription price: free. • Electronic access: <http://isedj.org/> • Contact person: Don Colton (editor@isedj.org)

2006 AITP Education Special Interest Group Board of Directors

Stuart A. Varden Pace University EDSIG President 2004		Paul M. Leidig Grand Valley State University EDSIG President 2005-2006		Don Colton Brigham Young Univ Hawaii Vice President 2005-2006	
Wendy Ceccucci Quinnipiac Univ Director 2006-07	Ronald I. Frank Pace University Secretary 2005-06	Kenneth A. Grant Ryerson University Director 2005-06	Albert L. Harris Appalachian St JISE Editor	Thomas N. Janicki Univ NC Wilmington Director 2006-07	
Jens O. Liegle Georgia State Univ Member Svcs 2006	Patricia Sendall Merrimack College Director 2006	Marcos Sivitanides Texas St San Marcos Chair ISECON 2006	Robert B. Sweeney U South Alabama Treasurer 2004-06	Gary Ury NW Missouri St Director 2006-07	

Information Systems Education Journal 2005-2006 Editorial and Review Board

Don Colton Brigham Young Univ Hawaii Editor		Thomas N. Janicki Univ of North Carolina Wilmington Associate Editor			
Samuel Abraham Siena Heights U	Tonda Bone Tarleton State U	Alan T. Burns DePaul University	Lucia Dettori DePaul University	Kenneth A. Grant Ryerson Univ	
Robert Grenier Saint Ambrose Univ	Owen P. Hall, Jr Pepperdine Univ	Jason B. Huett Univ W Georgia	James Lawler Pace University	Terri L. Lenox Westminster Coll	
Jens O. Liegle Georgia State U	Denise R. McGinnis Mesa State College	Therese D. O'Neil Indiana Univ PA	Alan R. Peslak Penn State Univ	Jack P. Russell Northwestern St U	
Jason H. Sharp Tarleton State U		Charles Woratschek Robert Morris Univ			

EDSIG activities include the publication of ISEDJ, the organization and execution of the annual ISECON conference held each fall, the publication of the Journal of Information Systems Education (JISE), and the designation and honoring of an IS Educator of the Year. • The Foundation for Information Technology Education has been the key sponsor of ISECON over the years. • The Association for Information Technology Professionals (AITP) provides the corporate umbrella under which EDSIG operates.

© Copyright 2006 EDSIG. In the spirit of academic freedom, permission is granted to make and distribute unlimited copies of this issue in its PDF or printed form, so long as the entire document is presented, and it is not modified in any substantial way.

An Alternative Testing Strategy for Advanced Programming Courses

Wendy Ceccucci, PhD
Information Systems Management, Quinnipiac University
Hamden, Connecticut 06518 USA
Wendy.Ceccucci@Quinnipiac.edu

Abstract

The traditional methodologies for evaluating student learning have been either a hand-written exam, usually in the form of multiple choice questions, or a take home program. These methods both have limitations which make it difficult to accurately measure a student's ability to apply the material learned to design and develop a program. An alternative testing strategy would be to let each student develop a take-home program exam for another student in the class. This evaluation strategy would allow students to uniquely apply the material learned. While developing their own programs, students would gain a better perspective of the limitations of their current knowledge.

Keywords: assessment, testing, exams, and advanced programming

1. INTRODUCTION

The two most common strategies for evaluating student knowledge in advanced programming courses, multiple choice questionnaires and take-home exams, have inherent limitations. Multiple choice exams are usually favored by faculty for a number of reasons. One reason is that multiple choice tests can be machine-graded, an advantage of special importance in large classes where volume grading is often required. Multiple choice tests also address the problem of time constraints. These tests enable instructors to ask questions that cover a wider range of material, and to ask more such questions than, say, essay tests (Bridgeman and Lewis, 1994, Walstad, and Becker, 1994). However, due to their basic nature, multiple choice questions do not test a student's ability to apply and develop a program. Additionally, there are also difficulties in traditional in-class exams due to time limitations since classes are usually either 50 or 75 minute segments which do not give students enough time to write a program of any substance.

One alternative to the in-class exam format is a take-home exam. This method allevi-

ates the time constraint, and allows the instructor to require more of a demonstration of skill by forcing students to write a full program. However, although instructors want to believe in the honesty and integrity of their students, realistically some students will communicate with one another and help each other with the exams. It is often difficult to measure the extent of sharing between students, especially if a program is correct and the variables are named differently.

In order to alleviate the problems found in the previous two methods, an alternative strategy is suggested. Students could be required to create take-home exams for another student in the class. By using this methodology, no student would have the same exam as another, thus eliminating the chance of information sharing. This approach has been found to be very successful in upper-level classes as both an evaluation and a learning tool.

2. METHODOLOGY

There are two primary parts to the exam process. The first part, worth fifty percent of the student's exam grade is the develop-

ment of a take-home program exam for another student. The second phase, worth the remaining fifty percent is for the student to complete the program exam that was written by another student.

Phase I – Develop a test

Students are first given a list of minimal requirements that they must include in the exam program. For example, the first exam covers chapters 1 - 4 of Bradley and Millspaugh's Advanced Programming Using Visual Basic .NET. The list of minimal requirements for the exam program would include:

- It must be a multi-tier application with a presentation tier and a database tier.
- The program should include two forms:

Form 1 - A drop down list should be populated from a field in the database. When the user selects an item, accompanying information should be displayed in labels. The label boxes should be bound to the data during run time

Form 2 - Create a program that displays data from tables that have a many-to-many relationship

- Must use data from one of the three databases given or create your own database. (This helps deter students from creating an example program that is too similar to the homework.)
- The connection string used to access the database must be generic.
- The program should be written so that it takes no more than 3 hours for an average student to complete.

Students are then given a few days to develop their test program. Once they have completed their test programs the students must then submit word documents containing their test instructions and program characteristics. They are required to turn in two disks containing the following:

1. The Professor's copy - A word document containing the directions and explanation of the test. A copy of the completed test program they created with the code and executable file. This disk should contain the name of the student.
2. A student's copy - A word document containing the directions and explanation

of the test. The executable file so the student taking the exam can have a working model to develop the program. There should be no programmable code on this disk. This disk should not have the student's name on it.

Phase 2 – Complete Take Home Program

Once the exams are written by the students and both disks are turned in, the tests are randomly distributed to the class. It is important that students are not told who wrote the exam they received in order to decrease the opportunity for assistance. The students are allowed to start their tests during class, which allows the instructor to address any problems that may arise. The students must then complete the take-home exams by the next class period. For a class size of 25 students, this creates 25 different exams. Each student has a different exam, so they are more likely to work independently.

Additional Rules

If a student is not able to create an exam, they will receive a zero for the first portion of the test and they are given either a take home exam created by the instructor or a test created by another student. Additionally, if a student's exam does not cover the minimal requirement or it is too similar to the homework or class work, appropriate points will be taken off.

3. LIMITATIONS

At the completion of the exam each student will have created two programs, making the grading process more time consuming than a multiple choice exam. This testing strategy would not be recommended for large classes. One possible solution to reduce the time required to grade would be to have students evaluate and grade the exams they are given to take. The students should document whether the program met the requirements and why they believe points should be taken off.

4. BENEFITS

There are many learning benefits in allowing students to create their own exams. First, students are forced to assess the material they have learned and to determine what is

most important. By evaluating the material in this manner, students will gain a better perspective. Secondly, students naturally want to apply what they are learning, so their test questions will demonstrate a creative application which is an important concept since the ability to create applications leads to a higher-level of understanding. In most programming class, students do not do a lot of writing. However, by writing the questions and instructions for the exam, the students will be developing their writing skills. Lastly, students will actually be taking two exams, the one they write and the one they receive.

5. CONCLUSION

Multiple choice exams are easy to deploy and can be easily graded, which lends itself to be the assessment strategy of choice for classes where volume grading is required. However, for smaller classes were alternative testing strategies are possible, enabling students to create their own exams for each other is a viable and successful mechanism. This strategy empowers the students by allowing them to creatively apply what they have learned and demonstrate their ability to develop and code programs.

6. REFERENCES

- Bradley, Julia and Anita Millsbaugh, Programming in Visual Basic .Net, McGraw-Hill, Irwin 2003
- Bridgeman, Brent and Charles Lewis (1994) "The Relationship of Essay and Multiple-Choice Scores with Grades in College Courses" Journal of Educational Measurement vol. 31, no. 1 (Spring), pp. 37-50.
- Walstad, William B. and William E. Becker (1994) "Achievement Differences on Multiple Choice and Essay Tests in Economics" American Economic Review, vol. 84, no. 2 (May), pp. 193-6.