AC 2008-442: THE IMPACT OF THE ALICE CURRICULUM ON COMMUNITY COLLEGE STUDENTS' ATTITUDES AND LEARNING WITH RESPECT TO COMPUTER SCIENCE

Ashlyn Hutchinson, Colorado School of Mines

Ashlyn Hutchinson (ashutchi@mines.edu) received her M.S. in Applied Mathematics, and is a Ph.D. candidate in the department of Mathematical and Computer Sciences at the Colorado School of Mines. Her area of focus is statistics, and her research interests inlcude assessment and biostatistics.

Barbara Moskal, Colorado School of Mines

Barbara M. Moskal (bmoskal@mines.edu <mailto:bmoskal@mines.edu) received her Ed.D. in Mathematics Education with a minor in Quantitative Research Methodology and her M.A. in Mathematics from the University of Pittsburgh. She is an Associate Professor in the Mathematical and Computer Sciences at the Colorado School of Mines. Her research interests include student assessment, K-12 outreach and equity issues. In 2000, she received a New Faculty Fellowship at the Frontiers in Education Conference and in 2006, she received the William Elgin Wickenden Award with her colleagues, Barbara Olds and Ronald Miller.

Stephen Cooper, Saint Joseph's University

Stephen Cooper is an Associate Professor of Computer Science at Saint Joseph's University. He taught previously at Rivier College, serving as Computer Science program director. He has also worked at IBM as a systems programmer. Dr. Cooper's research interests lie in the semantics of programming languages as well as in program visualization. He has been the principal investigator for several National Science Foundation and private grants, and serves as the college representative to the Computer Science Teacher's Association Board. (email: scooper@sju.edu)

Wanda Dann, Ithaca College

Dr. Wanda Dann is the director of the Alice team at Carnegie Mellon University. Her research has encompassed program visualization and object-oriented and event-driven programming. Dr. Dann has provided leadership in the international computer science education community, serving as SIGCSE 2004 Program co-Chair and SIGCSE 2005 Symposium co-Chair. She is now a member of the SIGCSE Board. (email: wpdann@andrew.cmu.edu)

Impact of the Alice Curriculum on Community College Students' Attitudes and Learning with Respect to Computer Science

Abstract

The Alice software is a three dimensional programming environment that is freely distributed through Carnegie Mellon University. Novice programmers can use the Alice software to create virtual worlds that consist of objects, or characters that interact on screen in a manner that is defined by the user. Users create these virtual worlds in Alice via a drag-drop editor, eliminating the frustration of syntax errors for beginning programmers. The Alice curriculum has been developed as a one-semester, introductory computer science curriculum that uses the Alice software to introduce students to programming. This curriculum was implemented and tested in three community colleges over three academic semesters. During this period, treatment data were collected in classrooms that implemented the Alice curriculum, and control data were collected in classrooms that used traditional programming languages as a first year computer science course. These data measured students' attitudes toward and learning with respect to programming. Based on the results of this investigation, students' attitudes with respect to computer science did not appear to be impacted either positively or negatively by the use of the Alice curriculum. However, students in the treatment classrooms did display greater gains in their programming knowledge when compared to students in the control classrooms. This improvement was more pronounced for men than for women, and this finding was consistent across the three semesters. This work was supported in part by the National Science Foundation (DUE-03020542).[1]

I. Introduction

For more than a decade, there has been a consistent decline in enrollment in computer science courses throughout the United States.[2,3] Additionally, students who decide to study computer science in college often leave this major prior to completion of their degrees.[4] Yet, the employment demand for qualified computer scientists and programmers is expected to continue to increase, resulting in a shortage of trained professionals in the field. This paper reports the results of a three year investigation that uses the Alice curriculum and software to introduce computer science concepts and programming to students who are attending community colleges. Attracting students to take an introductory computer programming course and retaining students once enrolled in the course (and ultimately in computing and computing-related careers) have been goals of the investigative team.

The Alice software, which was developed by Stage III at Carnegie Mellon University under the direction of Dr. Randy Pausch, is a three dimensional programming environment that is freely distributed at www.alice.org.[5] Novice programmers can use the Alice software to create virtual worlds that consist of objects, or characters that interact on screen in a manner that is defined by the user. Users create these virtual worlds in Alice via a drag-drop editor, eliminating the frustration of syntax errors for beginning programmers. The Alice software appeals to students who have been raised in a multi-media culture with animated movies and video games through its design. The Alice software supports students as they animate their own stories on screen. The

Alice curriculum has been developed as a one-semester, introductory computer science curriculum that uses the Alice software to introduce students to programming concepts. The appeal of the design of the Alice software is expected to support an appealing first semester introductory computer science curriculum. For further information regarding the Alice curriculum, textbook, and implementation options see [6][7].[6,7]

This paper discusses the results of the three year assessment efforts of the Java-Based Animation in Building viRtual Worlds for Object-oriented programming in Community colleges (JABRWOC) project which uses the Alice software and curriculum for teaching introductory computer science at community colleges.[1] The research questions that guide this investigation are as follows:

1. Does exposure to the Alice curriculum support students in developing the conceptual knowledge that is expected to result from introductory programming courses?
2. Does exposure to the Alice curriculum improve students' attitudes toward the field of computer science?
3. Does the appeal and effectiveness of the Alice curriculum differ for male and female students?

The research presented here was partially supported by the National Science Foundation (NSF, DUE-03020542).[1] The opinions and interpretations presented in this paper are that of the authors and are not necessarily reflective of or supported by the NSF.

II. Methods

The Alice curriculum was implemented at three community colleges and data was collected from students in both treatment (introductory computing course that used the Alice Curriculum) and control classrooms (the respective schools' original introductory computing course). The project was funded as a three year study. The first year was devoted to instrument development and a pilot study. The following three semesters, summative data was collected that investigators used to examine the effectiveness of the curriculum. This section describes the Alice software and curriculum, the participating classrooms and the measurement instruments used.

A. Alice Software and Curriculum

The Alice software was originally designed as an easy-to-use virtual reality simulator.[5] Toward this goal, it contains a drag-and-drop editor that does not permit syntax errors. In theory, this design can be used to teach students the algorithmic thinking of programming before learning the formal syntax of a programming language. If a program that uses the Alice software does not execute in the anticipated manner, the cause is the sequencing of the commands not the syntax. The abstract concept of an object, which is often difficult for students to grasp, is made concrete in the Alice environment through animations of people, animals and other physical objects that are displayed and manipulated. The Alice curriculum is designed to build on the features of the Alice software for instructional purposes. Through the Alice curriculum, students are taught the logic of programming without the burden of learning the syntax of programming. In theory, this reduces the cognitive load placed on students as they learn the logic of programming. A second

course is expected to follow the Alice course which introduces the students to the formal syntax of programming.

B. Treatment and Control Classrooms

The three community colleges which participated in this study were: Camden County College (CCC), Community College of Philadelphia (CCP), and Tompkins Cortland Community College (TC3).[7] As is indicated in Table 1, two of the participating schools had control classrooms; the third did not. The control classroom used the curriculum that had been in place for introductory computer science courses prior to this project. All of the courses impacted by the Alice curriculum were first year, introductory computer science courses.

CCC integrated the Alice curriculum into a semester long course, Fundamentals in Programming. The chosen control course was Computer Programming/QBASIC. This was considered the most appropriate control class for CCC, as it covers many of the same concepts as the Alice course. At CCP, two courses integrated the Alice curriculum over a 5 week period: PC Applications and Introduction to Programming. PC Applications also had control classes in a number of sections; it was the same course for treatment and control students, minus the five week period in which the Alice curriculum was implemented in treatment classrooms. Introduction to Programming had no control sections. TC3 utilized the Alice curriculum in two courses for five week periods: Introduction to Computer Programming and Introduction to Computer Information Systems. TC3 had no appropriate class from which to collect control data.

Table 1. Treatment and Control Courses

	Treatment	Control
CCC	Fundamentals in Programming	Computer Programming/QBASIC
CCP	PC Applications	PC Applications
	Introduction to Programming	N/A
TC3	Introduction to Computer Programming	N/A
	Introduction to Computer Information Systems	N/A

B. Instruments

The focus of this project's assessment efforts for the academic year 2003-2004 was formative in nature. During this period, instruments were selected, developed, and validated to be aligned with project goals. During the fall semester, several literature reviews were completed and a number of assessment instruments were examined. Based on this work, a Demographic Survey, a Concepts Exam, and an Attitude Survey were developed and/or chosen for the project. These instruments were then piloted in classrooms at the three community colleges. Since this was a formative phase, the primary purpose for this analysis was to determine what changes needed to be made to the assessment procedures in order to improve future implementations of this project. A number of changes were made to the instruments, as well as to the administration policies. For full details of the changes made based on the formative assessment, please see Hutchinson, et al.[8] Copies of all instruments are available via request from the first author.

The Demographic Survey collects background information on the participating students and supports the interpretation of project results. This instrument was administered to both treatment and control groups at the start of the semester. The collected data included: gender, ethnicity, community college attended, the course enrolled, and which section of control or treatment, and semester course was completed. Attempts were also made to determine the level of previous experience students had with computing; however, due to incomplete responses on the part of students, this data could not be used.

A Concepts Exam was created with the purpose of measuring how effective the Alice curriculum is for supporting the following student learning outcomes: 1) students' abilities to program with the Alice software, and 2) students' knowledge of basic computer programming concepts common to introductory programming courses. The exam format is multiple-choice, which eliminates the concern of inter-rater reliability. To ensure that the project implemented a valid instrument, several measures were taken. First, a programming novice completed the exam and the results were analyzed to determine if any questions could be easily answered without a background in the subject matter. Any questions deemed unsuitable were rewritten using guidelines for writing appropriate multiple choice questions. Next, a set of computer science experts reviewed the exam to evaluate the content that was being assessed. For complete details, please see Hutchinson, et al. [8] The first set of questions, henceforth known as *Basic Concepts,* assesses student's knowledge of computer science that is platform independent. In other words, this portion of the assessment examines basic computer science concepts that remain the same across different programming languages. The second portion of the exam, henceforth known as *Alice Specific*, asks questions that deal specifically with code and situations from the Alice software.

Both treatment and control students completed the Basic Concepts portion of the exam; only treatment students completed the Alice Specific portion of the exam. Both portions of the Concepts Exam were administered as pre and post instruments. Students in the treatment and control groups completed the pre exam at the beginning of their instruction with programming, and the post exam at the end. Both pre and post versions are identical. Each component of the exam is graded separately, and the score is determined by taking the number of correct answers out of the total questions in the instrument. The Alice Specific portion of this exam was only administered to treatment students in the fall 2004 and spring 2005. The decision not to administer this instrument in the final semester of summative data collection (fall 2005) was made by the investigators with the purpose of reducing the amount of class time required for evaluation.

The project investigators were also interested in determining whether using the Alice Curriculum would improve students' attitudes toward computer science. Therefore, a measure of students' attitudes was necessary and the Loyd-Gressard Attitude Survey was administered to both treatment and control students.[9] This survey uses a Likert rating scale with selected responses of Strongly Disagree, Disagree, Agree and Strongly Agree For analysis purposes, student responses were recoded numerically such that a higher score indicates a more positive attitude. After numerical coding, the range of scores was zero to three.

As with the Concepts Exam, the Attitude Survey is a pre and post instrument. Both treatment and control students completed the Attitude Survey at the beginning and end of their instruction with programming.

C. Data Collection and Analysis Procedures

All instruments were administered online through an online survey hosting site www.createsurvey.com.[10] Each data set was checked against the valid identification numbers for students who signed a project participation consent form. The data was then analyzed using the statistical computer package MINITAB.

To support the analysis process, the set of courses was divided into defined groups and sub-categories as described in Table 2. Although each course that implemented the Alice curriculum was computing-related, the courses were dissimilar enough that it was necessary to separate and examine the courses in isolation. In the case of the PC Applications course, students were enrolled in a variety of majors. For all other courses, students were identified primarily as computing-related majors, though not necessarily computer science majors.

For controls groups, the number of responses was too small to divide into sub-categories. Therefore, this table only reflects subdivisions that were used for the treatment groups. It is noted that for the final semester of summative data collection, the PC Applications course implemented two different forms of the Alice curriculum. The A sections used the curriculum described above, and was implemented during all examined semesters. The B sections used an alternate curriculum designed by and for CCP. This alternative curriculum was only implemented in the final semester. A factor that could not be controlled were the instructors of the treatment and control courses. In some instances, instructors taught both the treatment and control sections and in others they did not. The evaluators were not provided with information concerning this factor.

Table 2. Population Subsets for Treatment Groups

Main Category	Sub-Category	Definition
Total Population	Women	Includes all participating students in the treatment group
	Men	
PC Applications A & B	Women	Includes all participating students at CCP in CIS 103
	Men	
Semester Long Alice Course – CCC	Women	Includes all participating students at CCC
	Men	
Five Week Alice Courses – CCP Only	Women	Includes all participating students at CCP in CIS 106
	Men	
Five Week Alice Courses – TC3 Only	Women	Includes all participating students at TC3
	Men	

III. Results

The sections that follow present the summative data analysis on a year to year basis over the course of this project. The data collection procedures were quantitative in nature; no qualitative

data was collected. Two statistical tests were used: t-tests and Analysis of Co-Variance (ANCOVA).

A. Analysis Based on t-tests

In order to determine whether change occurred with respect to attitudes or learning of computer science concepts, two-sample t-tests were completed within each semester and across schools. Additional tests were performed within each gender across schools. Although both the attitudes and content instruments were administered in pre and post tests forms, the number of students consistently completing both pre and post versions of an assessment instrument was low. This was, in part, due to the high drop out rates of community colleges. This coupled with the desire to examine the data categorized by school, course and gender, resulted in the decision to retain as much data as possible by using unpaired t-tests. Therefore, the pre and post sets of data were treated as independent. This is a faulty assumption, given that this data set is neither completely dependent nor independent. Therefore, conclusions must be made with caution. Investigators also checked normality assumptions before testing the data. Non-normal data is noted in the tables and no additional analyses were completed on non-normal data.

A two-sample one-tailed t-test was completed on the pre and post means for both treatment and control groups to determine whether the pre and post scores differed significantly. If the mean of the post was greater than the mean of the pre, the following hypothesis was examined: $H_0 : \mu_{\mathrm{Pr}e} = \mu_{Post}$ versus $H_1 : \mu_{\mathrm{Pr}e} < \mu_{Post}$. If the mean of the post was less than the pre, the alternate hypothesis $H_1 : \mu_{\mathrm{Pr}e} > \mu_{Post}$ was tested. All tests were run for $\alpha = .05$ and the results are displayed in Table 3. For comparison purposes, only courses that had control groups (as indicated in Table 1) were included in this analysis. As Table 3 suggests, no significant differences were found within the control groups; however, significant differences were consistently found for the treatment groups.

Table 3. Comparison of Courses Implementing Treatment and Control Sections

				Concepts n-value		p-value: Concepts	Attitude n-value		
Semester	Course	Group	Subcategory	Pre	Post		Pre	Post	p-value: Attitude Survey
Fall 2004	PC Applications (CCP)	Treatment	Total Population	127	33	0.040*	138	48	0.341
			Women Only	76	17	0.040*	85	27	0.178
			Men Only	51	16	0.290	53	21	0.246
		Control	Total Population	79	18	0.051	127	33	0.983
			Women Only	34	8	0.084	70	11	0.338
			Men Only	45	10	0.168	57	22	0.200
Spring 2005	Semester Long Programming Course (CCC)	Treatment	Total Population	54	23	0.000*	79	25	0.158
			Women Only	16	6	0.000*	19	6	0.353
			Men Only	38	17	0.000*	60	19	0.273
		Control	Total Population	46	26	0.203	43	28	0.171
			Women Only	9	4	0.031*	6	5	0.426
			Men Only	37	22	0.871	37	23	***

*indicates significance at $\alpha = 0.05$

*** indicates non-normally distributed data

Table 3. (continued)

Fall 2005	PC Applications (CCP)	Treatment (A sections only)	Total Population	187	94	0.000*	272	98	0.007*
			Women Only	131	66	0.000*	176	63	0.010*
			Men Only	56	28	0.067	96	35	0.159
		Control	Total Population	44	18	0.414	57	25	0.886
			Women Only	32	11	0.805	42	15	***
			Men Only	12	7	0.292	15	10	0.494
	Semester Long Programming Course (CCC)	Treatment	Total Population	49	34	0.000*	45	34	0.527
			Women Only	8	5	0.004*	7	5	0.310
			Men Only	41	29	***	38	29	0.747
		Control	Total Population	55	43	0.071	57	44	0.985
			Women Only	10	9	0.417	10	9	0.955
			Men Only	45	34	0.066	47	35	0.486

*indicates significance at $\alpha = 0.05$

*** indicates non-normally distributed data

The next analysis examines these same differences but divided by the categories displayed in Table 1. This analysis only includes treatment classrooms because the sample size for the control groups was too small to support such an analysis. Additionally, since no comparisons are being made to control groups, all treatment groups were included in this analysis, even those that did not have a control group. Additionally, this analysis includes the results of a t-test on the pre and post scores of the Alice Specific content exam, which was only completed by the treatment group. The results by semester are displayed in Table 4. As this table indicates, the impact of the Alice curriculum on students' knowledge did differ based on the course in which the curriculum was included.

Table 4. Treatment Group: Concepts Exam and Attitude Survey t-test Results

Semester		p-value: Basic Concepts	p-value: Alice Specific	p-value: Attitude Survey
Fall 2004	PC Applications	0.040*	0.132	0.341
	PC Applications – Women	0.040*	0.283	0.178
	PC Applications – Men	0.290	***	0.246
	Semester Long Alice	0.000*	0.000*	0.206
	Semester Long Alice – Women	0.000*	***	0.328
	Semester Long Alice – Men	0.000*	0.000*	0.235
	Five Week Alice – CCP & TC3	0.000*	0.000*	0.418
	Five Week Alice – CCP & TC3 – Women	0.001*	***	0.355
	Five Week Alice – CCP & TC3 – Men	0.000*	0.000*	0.489
	Five Week Alice – TC3	0.000*	0.000*	0.268
	Five Week Alice – TC3 Women	0.000*	0.044*	0.151
	Five Week Alice – TC3 Men	0.0000*	0.000*	0.416
Spring 2005	PC Applications	0.175	***	0.210
	PC Applications – Women	0.406	0.202	0.174
	PC Applications – Men	0.125	***	0.880
	Semester Long Alice	0.000*	0.000*	0.158
	Semester Long Alice – Women	0.000*	0.000*	0.353
	Semester Long Alice – Men	0.000*	0.000*	0.273
	Five Week Alice – CCP	0.176	0.309	***
	Five Week Alice – CCP – Women	0.808	0.236	***
	Five Week Alice – CCP – Men	0.133	***	0.267
	Five Week Alice – TC3	0.000*	0.000*	0.476
	Five Week Alice – TC3 Women	0.000*	0.018*	0.566
	Five Week Alice – TC3 Men	0.000*	0.000*	0.413
Fall 2005	PC Applications A	0.000*	---	0.007*
	PC Applications A– Women	0.000*	---	0.010*
	PC Applications A– Men	0.067	---	0.159
	PC Applications B	0.455	---	0.242
	PC Applications B – Women	0.422	---	0.394
	PC Applications B – Men	0.437	---	0.264
	Semester Long Alice	0.000*	---	0.527
	Semester Long Alice – Women	0.004*	---	0.310
	Semester Long Alice – Men	***	---	0.747
	Five Week Alice – TC3	***	---	0.295
	Five Week Alice – TC3 Women	0.095	---	0.077
	Five Week Alice – TC3 Men	0.003*	---	0.851

*indicates significance at $\alpha = 0.05$

*** indicates non-normally distributed data

---indicates instrument was not administered

B. Analysis Based on ANCOVA

An Analysis of Covariance (ANCOVA) was performed within semesters across treatment data to determine if changes in attitudes and/or knowledge of computer science differed among gender. For this test, the following hypotheses were examined:

$$H_0 : \mu_{Male\_Post\_Scores} = \mu_{Female\_Post\_Scores}$$

versus

$$H_1 : \mu_{Male\_Post\_Scores} > \mu_{Female\_Post\_Scores}$$

The ANCOVA test requires paired data, that is, data from students who completed both a pre and a post version of an assessment instrument. The need for paired data significantly decreased the size of the sample that could be tested, and is indicated in Table 5 and Table 6. All tests were run for $\alpha = .05$. Significant differences were found between males and females, each semester the instruments were administered, for the Basic Concepts and Alice Specific portions of the Concepts Exam. With respect to attitudes, there were no significant differences between male and female responses from pre to post assessment.

Table 5. F-Value Results of Concepts Exam ANCOVA

Semester	n-value		Concepts Exam: Basic Concepts	Concepts Exam: Alice Specific
	Male	Female		
Fall 2004	108	42	5.003*	10.1103*
Spring 2005	44	31	11.63*	8.66*
Fall 2005	66	85	17.884*	---

*indicates significance at $\alpha = 0.05$
---indicates instrument was not administered

Table 6. F-Value Results of Attitude ANCOVA

Semester	n-value		Attitude
	Male	Female	
Fall 2004	102	52	0.373
Spring 2005	41	31	2.455
Fall 2005	76	85	1.62

*indicates significance at $\alpha = 0.05$
---indicates instrument was not administered

IV. Summary

Figure 1 provides a summary of the results of this investigation. As is indicated, treatment students displayed significantly greater mean post-test scores than pre-test scores on the Basic Concepts portion of the Concepts Exam within every semester that was analyzed. This was not found to be true for the control groups. Control classrooms did not display a statistically

significant difference between pre and post administration of this assessment for any of the examined semesters. This suggests that the use of the Alice Curriculum probably did have a positive impact on students' conceptual knowledge in computer science and this impact was greater than that which was witnessed in the traditional classrooms. Additionally, there was evidence to suggest that the nature of the given course and how the Alice curriculum was embedded in that course also had an impact on the extent of student learning. When investigating the differences in impact between men and women, the results indicate that on a semester-by-semester basis, males displayed larger differences between pre and post over the course of a semester than did females. This is consistent with previous research that has found a general trend with men make greater measurable learning gains than women in their initial computer science courses.[4] This interpretation must be made with caution, given that unpaired analyses were used in this investigation for data that was potentially dependent.

Figure 1. Summary of Results

- Across all semesters, the majority of treatment students displayed higher post-test scores than pre-test scores on the Basic Concepts exam; across all semesters, the majority of control students displayed no difference between pre and post-test scores on this same exam.
- The design of the course and how the Alice curriculum was embedded into the courses appeared to have an impact on student knowledge differences.
- The majority of treatment and control students across all semesters displayed no differences in their attitudes towards computer science and programming.
- In treatment classrooms for all semesters, men have more pronounced differences in knowledge than women.
- In treatment classrooms for all semesters, men and women do not display differing attitudes towards computer science and programming.

With respect to attitudes, there is no evidence to suggest that the use of the treatment or the control curriculum has an impact on students' attitudes in computer science. This, however, could be the result of the use of an outdated survey. Efforts are currently underway to develop an updated computer science attitude survey, which can be used to measure attitude changes in future studies.[11]

As is the case with all educational research, the results presented here should not be generalized beyond the participating population. Furthermore, these results are dependent upon the instruments and analyses used. Had different instruments or analyses been selected, the outcomes may have been different.

Acknowledgements

This research is supported by the National Science Foundation (NSF-DUE-0302542) and is part of the on-going efforts of the Center for Assessment of Science, Technology, Engineering and Mathematics at the Colorado School of Mines (see http://www.mines.edu/research/ca-stem/).

References

1. Cooper, S., Dann, W., & Moskal, B. *Java-Based Animation in Building viRtual Worlds for Object-oriented programming in Community colleges*. NSF-DUE-0302542.

2. Walker, Leslie., "Recognize Me?", *The Washington Post Online,* accessed 2006, http://www.washingtonpost.com/wp-dyn/content/article/2006/06/28/AR2006062802209.html?referrer=email&referrer=email&referrer=email.

3. Computing Research Association, *2004-2005 CRA Taulbee Survey,* accessed 2006, http://www.cra.org/CRN/articles/may06/tables.9to16.html.

4. Treu, K., Skinner, A. (2002). "Ten Suggestions for a Gender Equitable CS Classroom". *SIGCSE Bulletin – Inroads Special Issue: Women and Computing,* 34(2), 165-167.

5. *Alice: An Educational Software that teaches students computer programming in a 3D environment*, http://www.alice.org, accessed 7 January 2008.

6. Dann, W., Cooper, S., and Pausch, R., *Learning to Program with Alice*, Prentice Hall, 2006.

7. Curricular Materials for Learning to Program with Alice: Please contact author.

8. Hutchinson, A., Moskal, B., Cooper, S., & Dann, W. *Formative Assessment: An Illustrative Example Using Alice.* 2005 ASEE Annual Conference & Exposition.

9. Loyd, B.H. and Gressard, C.P. Computer Attitude Scale. *Journal of Computing Research,* 15(3), 241-259, 1996.

10. Online Survey Hosting, *CreateSurvey*, in use Fall 2003 through Summer 2006, www.createsurvey.com.

11. Hoegh, A. & Moskal, B._ (to appear 2008). *Examining engineering students' attitudes toward computer science*. Paper accepted for proceedings of the annual meeting of the American Society for Engineering Education,