Association for the Advancement of Computing In Education Journal, 15(1)

Kalman, H.K. & Ellis, M.L. (2007). Improving undergraduate computer instruction: Experiments and strategies, *AACE Journal, 15*(1), 23-38.

Improving Undergraduate Computer Instruction: Experiments and Strategies

HOWARD K. KALMAN

Ithaca College
Ithaca, NY USA
hkalman@worldnet.att.net

MAUREEN L. ELLIS

East Carolina University
Greenville, NC USA
ellism@ecu.edu

Today, undergraduate students enter college with increasingly more sophisticated computer skills compared to their counterparts of 20 years ago. However, many instructors are still using traditional instructional strategies to teach this new generation. This research study discusses a number of strategies that were employed to teach a college-level introductory computer literacy class. The strategies included a hands-on computer skills assessment, the introduction of a flexible attendance policy, elimination of the required textbook, and the use of learning teams. We explored alternative teaching methodologies in an effort to close the gap between classroom practice and real-world application and improve student learning.

Over the past 20 years, essentially all undergraduate education programs have replaced their state-mandated audio-visual course with a required introductory computer course. During the 1990s, most college students had little or no experience using computer software. Today, however, most incoming college students have take an introductory computer literacy course in middle school or high school or are self-taught (Johnson, 2002). As a result, college freshmen enter technology courses with increasingly sophisticated technology skills. Students' technology skills, however, vary widely: some students can barely insert a diskette while others are fluent in word processing and spreadsheet software, whereas still others can write computer programs and troubleshoot network problems. Educational computer instructors now face an increasingly difficult pedagogical challenge to accommodate both novice and advanced students in the same required introductory computer course. Teaching to the "middle" is unsatisfactory: beginners struggle and become overwhelmed while advanced students quickly become bored and unchallenged. This article describes experimental instructional strategies used to improve pedagogy and accommodate heterogeneous student populations in an introductory computer course.

We will begin by describing how the concept of computer literacy has evolved, then we will review instructional strategies employed in computer literacy courses to accommodate heterogeneous learners, next we will describe the existing course and the course redesign, and lastly, we will discuss the initial results of our redesign.

EVOLUTION OF COMPUTER LITERACY

The definition of computer literacy has evolved over the past 20 years. In the 20th Century, computer literacy courses focused on acquiring conceptual knowledge about how computers worked and gaining technical skills in using standard computer applications, such as word processing, electronic spreadsheets, and e-mail (Computer Science and Telecommunications Board, 1999). Today, the emphasis has shifted away from learning software features, which we believe students can learn independently, and more toward using computer tools to accomplish real-world tasks. Learners must be able to integrate their knowledge of technology skills, visual literacy, analytical skills, and critical thinking skills to solve complex problems. At the K-12 level, technology competencies are reflected in the National Educational Technology Standards for Students (International Society for Technology in Education, 2000). Similar competencies have also been identified by the U.S. Department of Labor as essential for our 21st Century workforce (Secretary's Commission on Acheiving Necessary Skills [SCANS], 1991). Yet, there are no standards across college-level introductory computer courses offered by various departments such as Computer Science, Education, and Business. Each department or school has different goals and objectives. In some cases, the emphasis continues to focus on ensuring that students can effectively use a word processing program to prepare academic papers in a specified format (such as APA or MLA). We believe that reinforcing skills that students have already mastered is not a good use of limited instructional time and may undermine college students' desire to learn.

TRADITIONAL INSTRUCTIONAL MODEL

The traditional demonstration-practice approach employed to teach introductory computer courses has remained essentially unchanged for the past 20 years. The instructor demonstrates how to perform a task step-by-step followed by the students practicing the behavior. Typically, once the students acquire basic proficiency in using one software application, the instructor moves on to the next software package. In a relatively short period of time, the students are exposed to a broad range of software applications, but none of these software programs is covered in depth. Although the traditional demonstration approach is efficient, students remain passive and dependent on the instructor for guidance (Hadley, 2002). More problematic, half the students' in our classes reported that the instructor's pacing of software demonstrations was too fast, while other students complained that the pacing was too slow—novices became frustrated, unable to keep up while advanced students became impatient or lethargic. More troubling, the demonstration approach is incongruent with current workplace demands in which employees are expected to acquire new technology skills independently through self-study (Goldsborough, 2003). The demonstration approach also severely limits opportunities for students to collaborate or participate in active learning strategies (Hadley).

STRATEGIES EMPLOYED TO BRIDGE THE GAP

Eventually, the skill gap of incoming freshman will narrow as state-mandated changes to K-12 computer literacy curriculums are implemented throughout the US (U.S. Department of Education, 2002). In the interim, however, colleges and universities have employed a wide range of strategies to accommodate their heterogeneous student populations. Some institutions allow incoming freshman to waive taking a required computer literacy course by successfully passing a skills assessment test. Other institutions provide quick-start workshops. While a “test out” option is useful for assessing technical skills, an easy-to-score test to assess the students’ ability to integrate technology concepts in performing real-world tasks, is less practical. The quality of commercial technical skills assessments also varies widely. For example, expensive computer-based systems simulate performing authentic tasks while less expensive assessments use multiple choice questions that ask the individual to recall inert knowledge (e.g., the student is asked to recall which pull-down menu to use to select a particular command). These types of assessments do not adequately test whether the students are capable of using computer software to accomplish meaningful tasks. Using a portfolio is another alternative, but is time-consuming for faculty to review.

The researchers found one teacher education program solved the problem by dividing their computer literacy requirements into three separate one-credit courses. The first course, which students have the option of testing out of, focuses on the acquisition of computer software skills. The second course focuses on appropriate integration of technology skills within classroom instruction. The third course involves implementing technology skills within an advanced field experience. The students acquire technology literacy skills within a broader context that focuses on direct application to the student’s major course of study. Our growing dissatisfaction with traditional instructional approaches led us to experiment with a diverse range of strategies to narrow the gap among our heterogeneous students while transforming our technology literacy curriculum. The next section will describe our existing course and the tactics we employed for redesign.

THE EXISTING COURSE

This research is based on data gathered during the past three years from a required introductory computer course taught at two research universities and one comprehensive college. These undergraduate courses were taught face-to-face and online**.** At one university, the course was designed for any undergraduate major; at the second university, the course was specifically designed for Education majors; at the comprehensive college, the course was designed for organizational communication majors. **S**tudents usually took the course during their freshman year, or the first semester they transferred into the program. At all institutions, the course was taught in a computer equipped classroom with one student per computer. Each course section consisted of 19-24 students. Data included instructor observations, students' informal anecdotal feedback, student evaluations, formal in-class debriefings at midterm and at the end of the semester, and a student skills assessment instrument.

The goal of the course was to introduce students to professional tasks, rather than emphasize mastery of computer application software. For example in word processing, students learned to create a school newsletter or employee biography incorporating technical elements such as footers, multiple columns, and graphics. Software applications taught included Microsoft Office (Word, PowerPoint, Excel, and Access) and Internet Explorer. In the comprehensive college's version of the course, students were also introduced to workplace tasks involving creating flowcharts using Microsoft Visio, Gantt charts using Microsoft Project, and desktop publishing using Adobe InDesign. Advanced students could also explore Adobe Illustrator and Adobe PageMaker. Technology concepts (e.g., computer hardware), ethics (e.g., copyright), and social concerns (e.g., digital divide) were also covered in all versions of the course. The required textbook included clear step-by-step instructions and was accompanied by relevant workplace-oriented practice exercises designed to appeal to 18-24 years olds. The textbook also prepared the students for the Microsoft Office Specialist Certification exam. While we thought the current course format and instructional tactics were working effectively, feedback from student evaluations, informal student discussions, and our own observations suggested otherwise. These signals led us to question the soundness of our existing pedagogical approach and initiated the course redesign.

COURSE REDESIGN

After considerable analysis and discussion among our colleagues, we identified a number of goals and strategies for the redesign, including: (a) individualize the course to better accommodate students' broad range of entry skills, (b) engage students in meaningful assignments, (c) promote student control and active learning, (d) encourage the use of alternative learning strategies (i.e., self-instructional tutorials), and (e) transform the instructor's role from a software demonstrator to a facilitator of inquiry.

Based upon our personal observations and the data we collected, we determined a need to close the gap between the methods we traditionally employed to teach computer technology and the natural approach that students appear to use to become computer literate. We rethought how we were currently teaching the course and how we could encourage the students to take control of their learning. Using the introductory computer course at the comprehensive college as our test site, we began redesigning the course by making three key changes: (a) a hands-on computer skills assessment was administered at the beginning of the course, (b) a flexible class attendance policy was implemented, and (c) the required textbook was eliminated.

Hands-on Computer Skills Assessment

During the first week of the semester, each student was required to take a hands-on computer skills assessment. The skills assessment replaced a self-reporting survey. The specific skills evaluated in the assessment are shown in Figure 1. After the assessment was scored, the student met with the instructor to review the results and develop an individualized learning plan. The individualized learning plan was shaped for each student based on the number and kind of skill deficiencies. For example, the typical student performed well on most Microsoft Word and PowerPoint tasks, but poorly on Microsoft Excel and Access. This student was advised to rectify skill deficiencies by self-study or by attending the hands-on demonstrations and then attend all class sessions on Microsoft Excel and Access. The instructor also suggested that the student tailor a number of assignments (e.g., the press release, information/training aid, and presentation assignment) to his or her career interest but use the instructor-prepared generic assignments for the spreadsheet and database assignments.

1.0	Microsoft Word
1.1	Create a table
1.2	Adjust the width of a table
1.3	Shade the cells in a table
1.4	Create a footer
1.5	Insert a symbol into the document
1.6	Insert automatic page numbering
1.7	Replace a word with a synonym
1.8	Create a custom bullet
1.9	Create a hanging indent
1.10	Create a mail merge document
2.0	Microsoft PowerPoint
2.1	Use a solid blue background for all slides
2.2	Color the font white
2.3	Create a horizontal line below the header in the master slide
2.4	Automatically number each slide
2.5	Insert clip art
2.6	Create a bullet list including sub-bullet items
2.7	Eliminate the slide number from the title slide
2.8	Add speaker notes to a slide
2.9	Print three slides per page
3.0	Microsoft Excel
3.1	Enter alphabetic and numerical data
3.2	Add columns and rows
3.3	Calculate the average (demonstrate using functions)
3.4	Format numbers using two digits to the right of the decimal and add “$” sign
3.5	Create a pie chart
4.0	Microsoft Access
4.1	Create a database
4.2	Create a primary key
4.3	Determine the field type
4.4	Enter records
4.5	Perform a query

Figure 1. List of specific skills evaluated in the computer skills assessment

For a student who performed well on most or all parts of the assessment, the instructor recommended that the student attend class demonstrations selectively, accelerate completion of the assignments, explore optional software, and spend more time working on the end-of-course portfolio. For the student who performed poorly on most parts of the assessment or who lacked confidence, the instructor recommended that the student attend all class sessions and use the instructor-prepared generic assignments throughout the course. Shortly after midterm, a second meeting was held with each student to review progress and make any necessary adjustments to the student's individual plan.

Flexible Class Attendance Policy

Approximately one-third of the class sessions were mandatory to ensure students attended to conceptual content and to introduce new, unfamiliar software programs. All remaining class sessions were optional. We expected that small groups of students would attend the software demonstrations and that students would make greater use of self-instructional learning resources such as print guides, videotapes, online tutorials, and coaching from a graduate assistant. We also expected that advanced students would work independently.

Required Textbook was Eliminated

We eliminated the required textbook because the students appeared to prefer using web resources and because once the technical skills were mastered, the textbook had little, if any, future value for the students. We provided a broader variety of learning resources such as videotapes, print-based tutorials with CD-ROMs, web-based tutorials, and hands-on workshops. The instructor also provided some supplemental guides and tip sheets, particularly for Microsoft Access.

INITIAL RESULTS

As expected, results from the computer skills assessment provided the instructor with powerful evidence that each student's skill set was highly variable, supporting the move to individualize the introductory computer course. Tables 1 through 4 illustrate how 13 students in one class section performed on each skill (Word, PowerPoint, Excel, and Access). The "x" indicates mastery of each skill.

Table 1
Student Mastery of Microsoft Word Skills

	Word									
	Table	Adjust width	Shade	Footer	Symbol	Insert pg#	Thesaurus	Custom bullet	Hanging indent	Mail merge
Student 1	x	x	x	x	x	x	x	x	x	
Student 2	x	x	x	x			x	x		
Student 3	x	x	x	x		x	x	x		
Student 4	x	x	x				x	x		
Student 5	x		x	x	x	x	x	x	x	
Student 6	x						x	x		
Student 7	x	x	x	x		x	x		x	
Student 8	x		x	x		x		x	x	
Student 9	x			x		x		x		
Student 10	x		x	x	x		x	x	x	
Student 11	x	x	x	x	x		x	x	x	
Student 12	x		x	x	x		x	x		
Student 13	x	x	x	x			x	x		

Table 2

Student Mastery of Microsoft PowerPoint Skills

	PowerPoint									
	Blue bkgrnd	White font	Footer	Customize master slide	Auto page #	ClipArt	Indent bullet	Omit # on title slide	Speaker note	Print
Student 1	x	x	x	x	x	x	x	x	x	x
Student 2	x	x			x	x	x	x		
Student 3	x	x				x	x	x		
Student 4										
Student 5	x	x	x	x		x	x	x		
Student 6	x	x					x			
Student 7	x	x	x		x	x				
Student 8	x	x			x	x				
Student 9										
Student 10	x	x			x	x	x	x		x
Student 11	x	x	x	x	x	x	x	x	x	x
Student 12	x	x				x	x	x		
Student 13	x	x				x	x			x

Table 3

Student Mastery of Microsoft Excel Skills

	Excel				
	Enter data	SUM	Function	$ Format	Pie chart
Student 1					
Student 2					
Student 3					
Student 4					
Student 5	x	x	x	x	x
Student 6	x			x	
Student 7					
Student 8					
Student 9	x	x	x		
Student 10	x	x	x	x	x
Student 11	x	x		x	x
Student 12					
Student 13					

Table 4

Student Mastery of Microsoft Access Skills

	Access					
	Access	Table	Primary key	Field type	Enter records	Query
Student 1						
Student 2						
Student 3						
Student 4						
Student 5	x	x	x	x	x	
Student 6						
Student 7						
Student 8						
Student 9						
Student 10	x	x	x	x	x	
Student 11						
Student 12						
Student 13						

While most students had mastered basic word processing skills (e.g., creating a table, creating a header/footer, and using the thesaurus feature), few had mastered intermediate word processing skills (e.g., creating a hanging indent and a mail merge document). We also observed that some students were proficient using Microsoft Excel and Access, yet were deficient in some basic word processing and presentation software skills.

During the assessment test, we observed many students attempting to figure out how to perform various tasks by using Microsoft's help feature. More interesting, the assessment test caused most students to reassess their computer competence more realistically. Several students commented during their initial meeting with the instructor that they thought they knew more than they did.

Instructional Strategies Employed

Flexible attendance strategy. Traditionally, attendance was mandatory since seating capacity in computer-equipped classrooms was limited and significant class time involved hands-on practice. Attendance was also mandated because students who did not attend class regularly usually performed poorly or failed. Yet, while we required class attendance, we also espoused that the student take control of his or her learning—our policy was incongruent with our belief. Initially, we made 50% of the class sessions mandatory and also required that students select a minimum of five optional classes to attend. Surprisingly, all students consistently attended all classes. The next semester, we mandated attendance for 40% of the class sessions. This time we observed three distinct patterns of attendance: (a) most novice and intermediate students, unsure of their skills, attended class regularly; (b) highly-advanced students attended only when required, but due to overconfidence, skipped some sessions which would have benefited them such as demonstrations on Microsoft Access; and (c) a few students lacked the self-discipline to regulate their attendance and would have benefited from a traditional mandatory attendance policy.

All students reported that they liked being given control of and responsibility for their attendance. While we believe that students cannot become self-directed learners when we mandate attendance, we are now considering creating an individualized attendance plan based on the results of the computer skills assessment and performance on early assignments.

Use of alternative instructional resources. Over the past several years we used a high-quality, full-color text from a major college publisher, complete with practice exercises directed toward college students. In place of the textbook, we encouraged the students to use the built-in help features of the software as well as human resources (e.g., classmates, friends, graduate assistants, tech support, and even parents)—mirroring how employees learn on-the-job. We were somewhat surprised that students seemed to prefer to attend traditional software demonstrations rather than engage in self-directed study. A number of novice students suggested that we make available an optional reference text.

Course assignments. With the exception of a short research paper, all assignments involved accomplishing realistic workplace tasks. Throughout the course, the students were asked to assume the role of an employee working for an organization—a difficult task for freshmen just beginning to explore career choices. Assignments included preparing a one-page employee biography, a press release, a training job aid, a process flowchart, an information presentation, a budget for a two-day conference, and a paper-based portfolio. For the portfolio, the students were required to include all of their previous assignments and create a resume, a business card, and three additional work samples. Students also had the option of including work from other classes or jobs. Organizing the portfolio forced the students to think about how to best present their competencies rather than merely show off their technical software skills. Students abandoned structuring their portfolios using a laundry list of software programs and organized their portfolios into individualized categories such as “internal communications,” “project management,” “graphics,” and “marketing.”

While it may seem premature to use professionally-oriented assignments with freshman, the challenge increased student motivation and forced them to begin to think more seriously about career interests. For students who already had a clear career direction, the assignments were straightforward and tailored to their specific career path. Other students did need more guidance from the instructor to prevent frustration. Additionally, the students were shown examples of finished projects so they had a better understanding of the instructor’s expectations for the assignments. The flexible structure of the assignments also accommodated those students whose career interests changed throughout the semester.

Use of learning teams. Another strategy we introduced in the fall of 2004 was the use of student learning teams. Our intent was to promote self-directed learning and to cultivate a learning community. Each team was responsible for delivering a workshop on a software program such as Microsoft Visio, Adobe Illustrator, or Microsoft Project. A job aid was provided to each team that described how to prepare their workshop and explained the rationale for the assignment. The rationale provided to the students is shown in Figure 2. Suggestions on the job aid included beginning the lesson with an overview of what the software is used for, showing examples of documents created with the software, and creating hands-on exercises based on real-world job tasks. To learn the software, each student team was provided print-based tutorials and CD-ROMs (e.g., Microsoft Step-by-Step). The teams were also provided a short list of specific technical skills, which they were expected to cover in their workshop. Additionally, they were required to meet with the instructor one to two weeks in advance of their workshop to review the context they had created for the lesson. The instructor also used the team meeting to reduce the potential for presentation anxiety and ensure that the team members were coordinating their individual sections. The instructor shifted his role to become a coach, and an instructional consultant rather than merely a resource for answering technical questions about the software. It should be noted that the use of learning teams and individualized meetings did place greater demands on the instructor's time; however, improvements in learner outcomes were worth the additional time investment.

Why we are doing this

First, in organizations, you will often be expected to learn software independently on your own. Second, the learning team approach provides experience working in teams and encourages you to help each other. Organizations often use teams to improve the quality and speed of projects. Third, the approach provides an opportunity to explore activities that might be part of your future career. Fourth, the approach ensures active rather than passive learning. The act of learning to teach encourages deeper understanding. Fifth, the approach recognizes that you are intelligent and capable and ensures that you take responsibility for your learning. Sixth, this is an experimental approach that we think will enhance your competence and confidence. Seventh, the team approach enables you to meet new people and have fun.

Figure 2. Excerpt from learning team guide sheet: Rationale for the assignment

So far, the results have been mixed. Students' limited experience presenting a lesson and their lack of confidence with the software reduced effectiveness of their presentation and their perception of success. Still, the students' reaction to the learning team assignment was positive. Most said they gained an understanding of the complexities of preparing and delivering a classroom lesson but suggested that we limit the presentation to 30 minutes and be very specific about the software features that they should cover. They also suggested that the instructor follow the lesson with additional practice exercises. Perhaps the most interesting finding is that the use of the learning teams has begun to transform the relationship between the students and the instructor from the traditional superior-subordinate relationship toward a learning partnership.

IMPLICATIONS AND CONCLUSIONS FOR THE STUDY

Since the earliest programmed textbook, introduced in 1657 by John Amos Comenius (Heinich, Molenda, Russell, & Smaldino, 1996), technology has played a valuable role in motivating students and supporting the learning process. Today, technology has increased the flexibility of instructional delivery options (e.g., online learning, computer-based tutorials, and blended learning) and has enabled individualizing instruction.

As computer literacy becomes mandatory in K-12 curriculums, the need for required introductory courses in higher education could become less necessary. The current generation of students entering higher education appears to assimilate technology naturally and transparently into their daily lives with little difficulty. Today's middle school students will arrive on college campuses with even greater technological competence. Instructors, however, are currently struggling to make sense of rapidly changing computer technology. Instructors must not only integrate technology into their teaching; they must rethink how they teach. As a result, today's digital divide may become less concerned with economic conditions and access to technology and more concerned with an individual's technical and conceptual competence. A new digital divide has emerged out of the students' abilities to naturally adapt new technological advancements and the instructors' inability to adjust their instructional modes and strategies to meet the ever-changing capabilities of increasingly sophisticated and technologically-savvy students. We look forward to continuing our experimentation and developing new ideas concerning best practices for computer instruction.

References

Computer Science and Telecommunications Board (1999). *Being fluent with information technology*. Washington, DC: National Academies Press.

Goldsborough, R. (2003). Taming technology fears. *Community College Week*, *28*, 7.

Hadley, N. (2002). An empowering pedagogy sequence for teaching technology. *Journal of Computing in Teacher Education, 19*, 1. Washington, DC: International Society for Technology in Education.

Heinich, R., Molenda, M., Russell, J.D., & Smaldino, S.E. (1996). *Instructional media and technology for learning* (5th ed.). Englewood Cliffs, NJ: Prentice Hall.

International Society for Technology in Education. (2000). *National educational technology standards.* Eugene, OR: International Society for Technology in Education.

Johnson, A. M. (2002). A new high school graduation experience requires increased staff computer literacy. *The Journal of Technology Studies, 28*(1), 78-81.

Secretary's Commission on Acheiving Necessary Skills, SCANS (1991, June). *What work requires of schools: A SCANS Report for America 2000*. Washington, DC: U.S. Department of Labor. Retrieved February 23, 2005 from http://wdr.doleta.gov/opr/fulltext/document.cfm?docn=6140

U.S. Department of Education (2002). *Technology in schools: Suggestions, tools and guidelines for assessing technology in elementary and secondary education.* Washington, DC: National Forum on Education Statistics. Retrieved February 23, 2005, from http://nces.ed.gov/pubs2003/2003313.pdf