Improving Computer Instruction: Experiments and Strategies

Howard K. Kalman
Ithaca College

Maureen L. Ellis
East Carolina University

Abstract

Today, undergraduate students enter college with increasingly more sophisticated computer skills compared to their counterparts of 20 years ago. However, instructors are still using traditional instructional strategies to teach this new generation. This research study examines strategies that instructors employ to teach introductory computer literacy classes in higher education. We explore alternative teaching methodologies in an effort to close the gap between classroom practice and real-world application.

Introduction

Over the past twenty years, essentially all undergraduate education programs have replaced their state-mandated audio-visual course with a required introductory computer course. During the 1990's, most college students had little or no experience using computer software, but today, most incoming college students have been exposed to an introductory computer literacy course in middle school or high school. As a result, college freshmen enter undergraduate technology courses with increasingly sophisticated technology skills. Instructors now face an increasingly difficult pedagogical challenge to accommodate both novice and advanced students in the same introductory computer course. Some students can barely insert a diskette while others are fluent in computer programming. Teaching to the "middle" is unsatisfactory: beginners struggle, and become overwhelmed while advanced students become bored, and unchallenged. This paper describes experimental instructional strategies to improve pedagogy and accommodate heterogeneous student populations in introductory computer courses.

We will begin by describing how 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 lastly, we will discuss the course redesign and initial results.

Computer Literacy

The definition of computer literacy has evolved over the past twenty 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 students must grasp the underlying principles of the technologies and understand how those principles relate to real-world tasks. Learners must be able to integrate their knowledge of technology skills, analytical skills, and critical thinking skills to solve complex problems. Yet, the emphasis of many introductory computer courses continues to focus on ensuring that students can effectively use a word processing program to prepare academic papers in a specified format. We believe that teaching skills which students have already mastered is not a good use of limited instructional time.

Outdated Instructional Methods

The traditional demonstration approach employed to teach introductory computer courses has remained essentially unchanged for the past twenty years. The instructor demonstrates how to perform a task step-by-step followed by the students practicing the behavior. Typically, once the students acquired basic proficiency in using one software application, the instructor would move on to the next software package. In a relatively short period of time, students were exposed to a broad range of software applications, but none of these software programs was covered in much depth. Although the traditional demonstration approach is efficient, students remain passive and dependent on the instructor for guidance

(Hadley, 2002). In addition, half the students' report that the instructor's pacing of software demonstrations is too fast, while other students complain that the pacing is too slow—novices become frustrated, unable to keep up while advanced students become impatient and 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). According to Hadley (2002), the demonstration approach also severely limits opportunities for students to collaborate or participate in active learning strategies.

Instructional strategies such as pairing advanced students with novice learners, allowing advanced students to proceed through the course material independently, or providing advanced students supplementary software is a "stop-gap" compromise at best. At the same time, the advantage of using a traditional "one size fits all" approach is that all students exit the introductory computer course with a consistent foundation and a common technical skill baseline.

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 rather than requiring students to take a computer literacy course. While a "test out" option is useful for assessing technical skills, for those instructors integrating the application of technology concepts into performing real-world tasks, an easy-to-score test is impractical. The quality of commercial technical skills assessments also varies widely. For example, expensive computer-based systems simulate students' performing authentic tasks while less expensive assessments use multiple choice questions that ask the individual to recall inert knowledge (e.g., ask 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 performing the application software skills.

Our research also found one teacher education program that divided their computer literacy requirements into three separate 1-credit courses. The first course (which students have the option of testing out of) focuses on acquisition of technical skills. The second 1-credit course focuses on the application of skills, and the third component involves the integration of computer and media skills within the teaching methods courses. The students acquire computer literacy skills within a broader context that focuses on direct application to the student's major course of study.

The Existing Course

This research is based on data gathered from a required introductory computer course taught at two research universities and one comprehensive college. The course is taught face-to-face and online for education and non-education undergraduate majors**. S**tudents usually take the course during their freshman year (or the first semester they transfer into the program). The course is taught in a computer equipped classroom with one student per computer and each course section consisted of 19-24 students.

The purpose of the course is to introduce students to professional tasks, not simply to master technical skills using computer software. For example, students learn to create a school newsletter or employee biography incorporating technical elements such as footers, multiple columns, and graphics. Software applications include Microsoft Office (Word, PowerPoint, Excel, and Access) and Internet Explorer. In the comprehensive college's version of the course, students are also introduced to workplace tasks involving Microsoft Visio, Microsoft Project, and Adobe InDesign; advanced students can also explore Adobe Illustrator and Adobe PageMaker. Technology concepts (e.g. computer hardware), ethics (e.g., copyright), and social concerns (e.g., digital divide) are also covered in all versions of the course. The required textbook includes clear step-by-step instructions and is accompanied by relevant workplace-oriented practice exercises designed to appeal to 18-24 years olds. The textbook also prepares the student for the Microsoft Office Specialist Certification exam.

Course Redesign

Based upon our observations and the data we collected, we determined a need to close the gap between the methods we traditionally employ to teach computer technology course and the natural

approach that students appear to use to become computer literate. We have begun to rethink how we teach the course and how we encourage students to take control of their learning. Using the introductory computer course at the comprehensive **college as our test site, w**e began to redesign the course by making three key changes: (1) a hands-on computer skills assessment test was administered at the beginning of the course, (2) a flexible class attendance policy was implemented, and (3) the required textbook was eliminated. During the second week of the semester, each student met with the instructor to review the results of the computer assessment and to develop an individualized learning plan. Shortly after midterm, a second meeting was held with each student to review progress and make any necessary adjustments. 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 learning resources such as print guides, videotapes, online tutorials, and coaching from a graduate assistant. We also expected that advanced students would work independently. We eliminated the required textbook because we found that once the technical skills were mastered, the textbook had little, if any, future value for the students.

Initial Results

As expected, results from the computer assessment test indicated that most students entered the course with varied skills in using Microsoft Word and PowerPoint. About 30% had some experience with Excel and less than 10% had experience with Microsoft Access. A small number of students (less than 10%) were deficient in basic word processing 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 that they thought they knew more than I 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 we believed that students who did not attend class regularly 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. Surprisingly, when we eliminated the attendance policy, most students (including advanced students) consistently attended class. Students like being given control of and responsibility for their attendance. Students cannot become self-directed learners when we mandate attendance.

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. We abandoned all textbooks because once the skills were acquired the textbook was no longer needed. We also noticed that our bookcases were filled with obsolete computer texts and manuals that we rarely, if ever, used for reference. Instead, we provide a broader variety of learning resources such as audiotapes, videotapes, print-based tutorials with CD-ROMs, web-based tutorials, and hands-on workshops. We also encouraged 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. At present, however, we are somewhat surprised that students seem to prefer to attend traditional software demonstrations rather than engage in self-directed study.

Use of Learning Teams

Another strategy we introduced fall 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 explained how to prepare the workshop. 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 that they were expected to cover in their workshop. Additionally, they were required to meet with the instructor 1-2 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 other difficulties. 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. So far the results have been mixed. Students' limited experience presenting a lesson reduced effectiveness and efficiency.

Discussion and Conclusions

As computer literacy becomes mandatory in K-12 curriculums, the need for required introductory courses in higher education could become unnecessary. The current generation of students entering higher education incorporates technology naturally and transparently into their daily life with little difficulty while instructors are struggling to make sense of rapidly changing computer technologies. Instructors must not only integrate technology into their teaching; instructors must rethink how they teach. As a result, today's digital divide is 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 technology-savvy students.

References

Being fluent with information technology, (1999). National Research Council, Committee on Information Technology Literacy, Computer Science and Telecommunications Board, Commission on Physical Sciences, Mathematics, and Applications. Washington, DC: National Academies Press.

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

Greene, B., Smerdon, B., Cronen, S., Lanahan, L., Anderson, J., Iannotti, N., et al. (2002) *Teachers tools for the 21st century: A report on teachers' use of technology.* NCES Report 2000102. Washington, DC: U.S. Department of Education, National Center for Education Statistics. Available online at http://nces.ed.gov/surveys/frss/publications/2000102.index.asp

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.

McMillan Culp, K., Honey, M., & Mandinach, E., (2003). A retrospective on twenty years of education technology policy. Washington, DC: U.S. Department of Education, Office of Educational Technology. Available online at http://www.ed.gov/rschstat/eval/tech/20years.pdf