Team Teaching an Advanced Computer Fluency Course: A Composite Perspective

Ned Kock
Lehigh University, Bethlehem, USA

nfk2@lehigh.edu

Robert Aiken, David Dalton, David Elesh, Anthony Ranere, Cheryl Sandas
Temple University, Philadelphia, USA

aiken@joda.cis.temple.edu   dalton@astro.ocis.temple.edu
delesh@thunder.ocis.temple.edu   ranere@vm.temple.edu
csandas@snowhite.cis.temple.edu

Abstract

This paper discusses the observations of six instructors who team-taught an advanced computer fluency course over a period of three years. The course exposed students to complex information technology applications, such as geographic information and molecular design systems, in specific professional domains. One of the main goals of the course was to give students a glimpse at real-world applications of information technology aimed at solving complex problems. In addition to providing personal observations we summarize some of the problems that were encountered and how we addressed them. Also, the result of analyzing some preliminary data is discussed. The goal is to assist other instructors who might be interested in designing/teaching a similar course.

Keywords: Computer fluency, Non-majors, Lifelong learning, Co-teaching, Cross-disciplinary teaching

Introduction

The authors of this paper developed and taught a course designed to provide students, not majoring in computing, the skills and confidence to go beyond the introductory computer literacy course. The objective was to prepare students to integrate IT seamlessly into the fabric of their specific disciplines (Aiken, Kock & Mandviwalla, 2000). The vehicle we chose was to incorporate in-depth case studies as the best approximation to apprenticeship learning. In brief, the goals are to give students the background and motivation to both feel comfortable with using technology as well as encourage them to want to stay current with technological deve lopment. An early perspective on teaching this course can be found in (Kock, Aiken & Sandas, 2002).

Planning for this project was underway for several years before the "Being Fluent with Information Technology" report appeared (Committee on Information Technology Literacy, 1999) however, in many ways our ideas and concerns are similar. . As has been noted elsewhere there is a recognized need and large demand for a course such as this (McEuen, 2001). Moreover, there are ongoing efforts in addition to ours to integrate Computer

Fluency into the curriculum, see for example http://www.cs.haverford.edu/Itfluency .

Principal objectives of our approach include establishing a framework to develop portable case studies, assessing the effectiveness of the course in achieving its stated goals, and disseminating our results as widely as possible. This is certainly consistent with the above mentioned report which states, "In summary, FIT individuals, those who know a starter set of IT skills, who understand the basic concepts on which IT is founded, and who have engaged in the higher-level thinking embodied in the intellectual capabilities, should use information technology confidently, should come to work ready to learn new business systems quickly and use them effectively, should be able to apply IT to personally relevant problems, and should be able to adapt to the inevitable change as IT evolves over their lifetime" (Committee on Information Technology Literacy, 1999, p. 5).

We hope that other colleagues interested in teaching a similar course can benefit from our experiences as described in the following sections. In the next section we discuss some of the key features of the course, followed by a summary of the views of the six different lecturers, an analysis of the strengths and weaknesses of the course, and finally a few concluding remarks and thoughts about the future of such a course.

Key Aspects of the Course

This is a one semester fourteen week course that consists of an Introductory week, six two week segments taught by six different instructors, and a final week devoted to demonstrating and discussing student projects. The basic content is encapsulated in three case study lectures and labs (two weeks each) taught by professors from Anthropology, Chemistry and Sociology; six weeks of information technology lectures and labs taught by a professor in CS and two in MIS, a weekly journal, and a final project. Although the foundation for achieving the aforementioned objectives is laid in the lectures and labs, they are constantly reinforced through the use of journals and a final project.

Journals allow the students to reflect on individual sections of the course as well as the course as a whole. By mandating journals, we gain insight into the cognitive processes of the students that we may not gain otherwise. For example one student remarked "The application of this type procedure will definitely be useful when interacting with almost any type of software or even with products that are not directly related to computers" [sic]. The requirement to generate a written record encourages students to reflect and connect the concepts presented. Students can discover how the IT with which they are interacting can be used within their own discipline/personal life in addition to how IT is used within the discipline of the case study presenter. For example one student noted that "Shift-share analysis and spread sheets can probably be used in almost any discipline in which large amounts of data have to organized and analyzed" [sic].

The final project provides students with an opportunity to demonstrate their knowledge of IT and synthesize the concepts covered throughout the course. Students can develop their own final project or choose from among the ones provided. All projects must be available via an organized web site. In addition, students must supply two progress reports during the semester and present their project web site to the class.

For example, one student chose to create a web site for a psychic business. This student used MS FrontPage to create his site, MS Excel to create the spreadsheets in his business plan, and MS Word to compose the written content of the business plan. In addition, he used the web to research psychic sites for content and prices to forecast his sales. He also used the web to assess the competition for this type of business. The end result was a working psychic web site and links to the mission statement, business plan, and contact information.

By having this framework for the final projects, students realized the following objectives: an understanding of basic IT concepts, experience in expressive and collaborative communication, and experience in using technology productivity tools, research tools, and problem solving and decision making tools. More details about the course can be found on the course website at http://ww2.cis.temple.edu/nsflll/. Table 1 provides an outline of the major components of the course.

Week 1	In Class: Review of course goals, organization, and historical overview of IT.
	Lab: Introduction to the Laboratory and Internet. Demonstration of the technology used in the course with a simple group project.
Week 2	In Class: Data, Representation, and Algorithms. Being Digital. The computer system as a universal simulator/modeler for solving complex problems.
	Lab: Continuation of lab 1 with focus on creating and sharing information on the Internet (word processor as an HTML editor) and discussing information (electronic bulletin board). Team modeling (drawing and shared workspace tool).
Week 3	In Class: Case Study 1: Deforestation in tropical America using archeological and paleoecological data from 9000 to 2000 BP (Before Present).
	Lab: Large scale record keeping (Enterprise and single user DBMS), visualization of location data (Geographic Information System). Demonstration of simple simulation models.
Week 4	In Class: Case Study 1: Continue with case study with focus on class project. Abstraction of study as an example of modeling long-term human behavior.
	Lab: Complete assignment. Other examples of visualization and modeling using GIS systems.
Week 5	In Class: The structure and evolution of computer systems, from chips to supercomputers. Rate of change, performance, and cost.
	Lab: Use of benchmarking tools and software to evaluate the systems in the lab. Modeling computer systems.
Week 6	In Class: Software: Components, architecture, development. How software is created, the existence and extinction of categories of software, and the marketing and packaging of software. Identifying the "right" technology for solving a problem.
	Lab: Examples [audio/video] showing that quantitative changes in performance have a qualitative impact on user satisfaction. Demonstration of statistical analysis problems, models, methods, and tools.
Week 7	In Class: Case Study 2: Occupational and age cohort consequences of the industrial transformation.
	Lab: Shift-share analysis using a spreadsheet. Each team selects alternative assumptions and hypotheses (some selections are predetermined, others are arrived at through collaborative brainstorming) and works on the case study.
Week 8	In Class: Case Study 2: Continue. Abstraction of study as an example of computer based analysis of data.
	Lab: Complete assignment and comparison of results.
Week 9	In Class: The structure and evolution of Data Communication and networks. The interaction of technological, economical, and political factors. Using communication technology for collaboration.
	Lab: Examples of Pull and Push communication models. Compare and test collaboration technologies (synchronous and asynchronous)

Week 10	In Class: Case Study 3: Exploring Structures of Organic Molecules by Computational Methods. Using laws of physics and appropriate computational methods to predict the structures and properties of simple molecules Lab: Calculations are performed with the aid of commercially available software (e.g. Alchemy, HyperChem, Spartan)
Week 11	In Class: Case Study 3: Continue. Comparison of energies of structures to predict physical properties which can be compared to those found in data bases of chemical properties Lab: Complete assignment associated with case study 3.
Week 12	In Class: Artificial Intelligence and Learning. Advanced applications including Intelligent Tutoring Systems. Lab: An expert system. Neural net application that shows simple learning. Example of data mining. Mechanical hypothesis formation.
Week 13	In Class: The economics, social implications, and future of IT. Lab: Demonstration of individual and team multimedia term reports (audio/video capture and editing). Critique other people's work (collaborative authoring and electronic bulletin board).
Week 14	In Class and in Lab: Conclusions, term reports, demonstrations, test.

Table 1: Major components of the course

Non-CIS Instructors Perceptions of the Course

Following are some of the observations of the Instructors from Anthropology, Chemistry and Sociology (n.b. all comments are quotes taken from brief reports prepared by each of the instructors):

"Computer fluency is an ill defined and moving target. On the one hand, the plethora of ever more sophisticated programs designed to meet objectives their creator(s) believed (commercially?) worthwhile is not static. Revisions as well as competing versions often abound. On the other hand, common themes can be found. One concept that can serve as a motif to increase computer fluency across disciplines is the use of computers to model real systems. That is, it should be made clear that some fragment of reality is to be modeled and then the way the model works both with and without the computer discussed. The computer is a tool used to bypass arduous, repetitive, and error prone routines and a new level of fluency, beyond keyboarding and routine use of Microsoft ® *et al.* programs can be learned." (Chemistry Professor)

"There seems to me to be two related goals in teaching a computer fluency course: (1) giving students hands-on experience with different software packages, and (2) showing students how different software packages (or classes of software; GIS, spreadsheets, statistical packages, etc.) could be useful in their own fields. The principal strength of this course, it seems to me, is the exposure it gives students to a variety of computer software, to actual applications of computer software in different fields and hands-on experience in using the software. The course would probably be more meaningful to students if it were directed towards their own fields or towards closely related fields: e.g. computer fluency in anthropology, computer fluency in biology; computer fluency in the social sciences, computer fluency in earth sciences. This would also allow for better integration of case studies to provide more continuity in the course." (Anthropology Professor)

"This diversity (of student backgrounds) and a corresponding–though not necessarily correlated–range of familiarity with technology made the course a continual instructional challenge. In fact, I overestimated their preparation, and I significantly revised my module twice in the five times I taught it, each time trying to create a more step-by-step process of learning the material. During one term, I also substituted an additional lab session for a lecture, and I think it materially contributed to the success of the term." (Sociology Professor) (n.b. this point was mentioned by all three of these instructors)

What We Learned

Among the lessons we learned, which in part are captured by the observations in the previous section, are the following – presented as issues and solutions:

Issue #1: Creating a unifying theme and common threads among the different modules.

Solution: With six different instructors it was difficult to accomplish the above. The solution seems to be to develop themes and practices which each instructor can build on. We organized the course around themes such as model building, user interface issues and implications of the use of advanced technology. We provided hands-on opportunities in the labs and attempted to have the students tie everything together in their final project.

Issue #2: Addressing administrative issues such as how to assign workload credit.

Solution: This was not an issue for us since we received funding to pay the instructors and a graduate teaching assistant. But, this will be an issue for others developing a similar course if they use several different instructors. Our suggestion is to have the course "team taught" by several faculty in the same department or college; perhaps three faculty who each teach a four week segment with one of them serving as the course coordinator. Each person would receive credit for the course every third semester – most likely the time when they act as the coordinator.

Issue #3: Making the material more relevant for the students.

Solution: Since our three non-CIS as well as the three CIS case studies were quite diverse this was not easy, but trying to show how each case study embodied "real life" features seems to have been achieved successfully. In conjunction with point #2 we suggest that faculty from similar disciplines team-teach the course. A key factor is to find knowledgeable computer users who are using computers in their research and are enthusiastic about developing a computer fluency course. Thus, variations of this course might be offered; perhaps one by faculty from several departments in the business school for business majors, another course taught by College of Education faculty for majors in the College of Education, etc.

Issue #4: Providing students a mechanism to tie together the disparate concepts and techniques they had learned.

Solution: Students did a final project that required them to use some of the software and techniques they had learned. They were also required to incorporate aspects using new software. The students found this piece of the course to be the most useful so in successive offerings of the course we put more emphasis on the project (both in terms of more free lab time as well as increasing its weight in the grade). One suggestion would be to have the case study lecturers develop final projects that allow students to pursue topics stemming from their cases. Additional discussion of this and related data from student evaluations can be found in Kock, Aiken and Sandas (2002).

Conclusions

Developing and teaching this course has been a difficult, but rewarding challenge. Being able to work closely with colleagues from different disciplines has been one of the most enjoyable aspects. In addition, all the students who have taken the course say they have enjoyed it and learned a lot. Many of them gave us suggestions that led to improvements in successive offerings.

More specifically a preliminary analysis of data collected from the students in connection with their perceptions of the course suggest that the case studies had a positive impact on how students who are not particularly interested in IT perceive IT in general, and how it can be applied to complex and specialized tasks. While the students generally perceived that they learned something from the case studies, the results also suggest that the case studies did not have a positive impact on the students' willingness to pursue an IT career or take other IT courses. One plausible explanation for this attitude is that subjecting non-majors to complex and specialized IT applications increases their "respect" for IT's potential, but does not give them confidence to become specialized IT users themselves. It is quite possible that the degree of technological and domain complexity in the case studies led the students to believe that using IT was too difficult for the tasks that they may have to accomplish in their chosen fields of study (Kock, Aiken & Sandas, 2002).

We certainly came away convinced that the central component of our approach – using a diverse set of case studies – was successful. Also, requiring the students to develop an independent project proved to be an even more important aspect than we originally realized. Others who developed a similar course have made the same observation – see, e.g., www.cs.washington.edu/education/courses/100/CurrentQtr/.

In addition, requiring the students to keep a weekly journal allowed us to discuss a number of timely social issues that were impacted by advanced technology. In retrospect we might have been able to make better use of these journals than we did by following them more closely during the semester.

In summary, this was a fun course to develop and teach, and students did find the course worthwhile and gained confidence in using new tools. However, it comes with a certain amount of "administrivia" that can become stumbling blocks; not the least of which is the assignment of course workload in a team-teaching environment. But we feel the course is worth the effort, and would be glad to work with other colleagues who might like to implement a similar course.

Acknowledgments

References

Aiken, R., Kock, N. & Mandviwalla, M. (2000). Fluency in Information Technology: A Second Course for non-CIS Majors. *Proceedings of SIGCSE* 2000. New York, NY: ACM Press, 280-284.

Committee on Information Technology Literacy (1999). *Being Fluent with Information Technology*. Washington, D.C.: National Academy Press.

Kock, N., Aiken, R. & Sandas, C. (2002). Using Complex IT in Specific Domains: Developing and Assessing a Course for Non-Majors. *IEEE Transactions on Education 45(1)*, 50-56.

McEuen, S.F. (2001). How Fluent with Information Technology Are Our Students. *EDUCAUSE QUARTERLY 4*, 8-17.

Biographies

Ned Kock is Director of the E-Collaboration Research Center and Class of 1961 Associate Professor, Lehigh University, Bethlehem, PA. He holds a PhD in information systems from the University of Waikato, New Zealand. He is interested in the impact of new IT teaching techniques and online learning tools. He is the author/co-author of four books, and has authored articles published in a number of journals including *Communications of the ACM; IEEE Transactions on: Education, Engineering Management, and Professional Communication; IFIP Education and Information Technologies, Journal of Information Technology Education*, and *Knowledge and Process Management*.

Robert Aiken is a Professor and Director of the Research Committee in the Computer and Information Sciences Department, Temple University. He is the (co-) author of three books as well as more than 80 articles in refereed journals and proceedings. His current research activity includes investigating the applicability of artificial intelligence models to education, developing collaborative learning systems and assessing the impact of technology in K-12 education. Dr. Aiken has consulted with UNESCO and OECD, and governmental organizations of several countries on developing and assessing graduate programs in Computer Science and strategic planning for the effective integration of computers into schools.

David Dalton is Professor of Chemistry at Temple University. He received his Ph.D. degree from UCLA. He has long been concerned with the communication of information through the signs and symbols (semiotics) used by the sciences to those not specifically trained in the disciplines for which they were invented.

David Elesh is Associate Professor of Sociology and Director of the Social Science Data Library at Temple University, Philadelphia, Pennsylvania. He holds a Ph.D. in Sociology from Columbia University in New York. An urban sociologist, his work primarily involves the study of changes in industrial structure and their consequences for inequality. Long involved with the application of IT technology to teaching and research, he is chair of the American Sociological Association's section on Communications and Information Technologies. He has co-authored one book and published numerous articles in sociological journals such as *Sociology of Education*, *Journal of Human Resources*, and *Social Science Research*.

Anthony Ranere is a Professor of Anthropology at Temple University. He received a Ph.D. in anthropology from the University of California at Davis. He has carried out archaeological research in North, Central and South America. His research interests include lithic technology, paleoecology, evolutionary theory and spatial analysis. Current projects involve computer simulation of population growth, deforestation, upland soil erosion and lowland sedimentation in Panama using GIS (Geographic Information Systems) software. He is the author or co-author of over 40 journal articles and book chapters.

Cheryl Sandas holds a M.S. in computer and information sciences from Temple University and a B.S. in anthropology and psychology from Carnegie Mellon University. She is interested in the impact of technology on education. Currently, she is working at GlaxoSmithKline.