

# TEACHING OBJECT-ORIENTED APPLICATIONS VIA DISTANCE EDUCATION

Dr. Alan R. Peslak, Penn State University, arp14@psu.edu

## ABSTRACT

*Due to the increasing popularity of distance education, it is important to understand the factors of distance education student success. The author reviews the literature and finds eight factors which have proven important – hands-on experience, course design, variety, clarity, feedback, rapport, technical support, and content. The author reviews his experience and methods describing his inclusion of these factors into his object-oriented applications course.*

**Keywords:** Distance education, object-oriented applications, UML

## INTRODUCTION AND BACKGROUND

Distance education as a method of delivery for higher education in the United States has grown significantly in popularity over the last several years. The US Department of Education [14] has prepared a comprehensive report which illustrates the status and importance of distance education. Their report suggests that distance education is a major force in higher education. Also noted is the importance of asynchronous Internet delivery of content as an overwhelmingly popular delivery mode. As of 2000-2001 there were 3,077,000 distance education enrollments estimated in 2 and 4 year degree institutions. Given the importance of distance education, it is essential that proper pedagogical methods be applied to maximize the educational value of this non-traditional process. This report is a review of successful pedagogical methods in providing quality distance education via the Internet. Specifically the article also reviews the lessons learned from teaching programming and the complexities of object-oriented applications via the Internet. The author provides a review of the literature and derives key principles of success for both distance education in general and object-oriented programming in particular. The report then reviews the author's implementation of the eight factors in an online distance education graduate course in object-oriented applications at a national institution of higher learning.

## REVIEW OF LITERATURE

### General Distance Education

An examination of the distance education literature suggested seven initial general factors which are critical to successful general delivery of higher educational content over the Internet. These concepts are presented below.

**Course design.** Course design is crucial. Eastmond [4] suggests that particularly in online delivery, proper design is essential to student success. Eastmond notes many advantages and features of Internet delivery such as collaboration, interaction, self-paced study and individualized experience but all these must be explicitly designed into the course content.

**Clarity.** The need for clarity of instructions is paramount. Hara and Kling [5] performed an ethnographic study of “distress” that web-based students had with distance education courses. One of the major distresses noted was ambiguous instructions. Without the opportunity for face to face clarification, it is especially critical for instructors to provide clear syllabi and assignment instructions to their students. Jegede, Taplin, Fan, May, Chan, and Yum [7] suggest that clear instructions can improve a student’s sense of control, a perception which correlates positively with online student success.

**Feedback.** Hillesheim [6] notes the unique nature of distance education where traditional classroom feedback is non-existent. Specific methods to provide fast, effective feedback are recommended including direct individual feedback, collective feedback, asynchronous discussions and monitoring these discussions.

**Rapport with students.** Piercy [10] found that development of a strong rapport with online students is difficult but essential to a “positive teaching and learning experience”.

**Technical support.** Wang [15] reviews the literature and finds technological support to be one of the most critical elements of distance education success even in 1994. These findings are shared recently by Piercy [10], Stidham and Frieden [12] and Soong, Chan, Chua, and Loh [11].

**Variety.** Alley and Jansak [2] note that “learning is unique to the individual”. The authors suggest that providing a variety of pedagogical methods allows online students to individualize their instruction to their own preferences. The student selects from various modes of instruction such as text, visual, activities, but variety of methods must be built into the curriculum.

**Hands-on experience.** Alley and Jansak [2] suggest that in order for humans to develop competencies, actual experience is necessary. For transfer of knowledge from short-term to long-term memory, experience is vital. The author suggests that Internet delivery can be successful but for some period of time; online students need to step away from the instructional site and actually perform the studied activities.

### **Programming and Object-Oriented Distance Education**

There are relatively few research articles that study teaching of programming via distance education. One of the studies was performed by Molstad [9] at Dakota State University. One of the missions of Dakota State is to provide delivery to all state residents. The school has therefore become a pioneer in development and delivery of distance higher education. The school faced a challenge with delivery of information technology literacy courses that focused on programming and computer applications. Some of the findings of the study suggested the following success factors:

- Clear syllabus and objectives
- Multiple modes for feedback
- Technology skills identified
- Clear communication channels

- Technology support including grading and information resources
- Good explanatory textbook

Results suggested that those with no programming background had difficulties understanding lab assignments. Molstad notes that those with some background had a much more favorable experience. This could suggest that teaching programming solely online without any experience or direct instruction may be difficult. Molstad did not attempt to address this shortcoming.

Thomas [13] also provides some lessons learned in her experiences teaching C++ programming online. She had a generally positive experience with students who had some prior experience with programming. Many of the suggestions of others are also found here including

- Necessity of a readable textbook
- Need for self-motivated students
- Proactive instructor with active dialogue
- Quick responses to questions and assignments quickly

McGill, Volet, and Hobbs [8] found the following factors which led to success in distance education computer programming courses: previous programming experience, entrance scores, estimated hours per week doing practical work, confidence in ability, and expectation levels and internal hard work. Abraham [1] found the several lessons learned as a result of teaching an online introductory Visual Basic course. He suggests the following are essential for success of teaching programming online:

- Timely feedback – programming instruction is best facilitated by immediate or as near as possible feedback. Programming students can frequently get frustrated by syntax and logic difficulties which can be difficult to find for a novice.
- Motivation to participate in group discussion – to promote active involvement.
- Understanding the importance of content delivery – structure delivery to involve students.
- Creativity in developing assessment, feedback, grading etc. – The unique environment requires multiple and creative modes.
- 

With regard to teaching object-oriented applications, Burton and Bruhn [3] strongly recommend the use of UML in teaching Object-Oriented Systems Analysis and Design. The authors see significant benefit in the ability of the students to visually represent the abstract concepts of OO. The authors suggest that the best way to learn OO and UML is to customize instructor developed models.

The results of the limited literature published on object-oriented programming and applications via distance education also suggest that the many of the same factors of success apply for OO applications as for any other distance education course. Molstad [9] emphasizes clarity, feedback, and technical skills. Abraham [1] notes the importance of feedback and rapport. Thomas [13] also notes the importance of feedback and communications. McGill, Volet, and Hobbs [8] also suggest rapport. The one additional factor which is mentioned by Molstad [9], and Thomas [13] is **content** in both the quality of the textbook and the content of assignments. This is the eighth factor for OO online success.

## **OBJECT-ORIENTED APPLICATIONS COURSE**

The author has taught ten distance graduate computer information systems courses over a three year period including software engineering, survey of programming languages, and database systems. In addition the author has taught an object-oriented applications course three times, once in 2003, once in 2004, and currently in 2005. In general the instructor has applied the first seven factors of distance education success in these courses. Particular attention was also paid to the eighth factor, content, which in this case is programming. The following subsections describe the inclusion of each key success factor in the author's object-oriented course.

**Course design.** The course design for an online course in object-oriented programming starts with a clearly developed syllabus. The syllabus must contain all the essential elements of the course. (My complete syllabus is available via email.) The objectives and approach should be clearly detailed and drive the overall design. The object-oriented application course was geared to not just teaching programming but also teaching the design tools and the OO paradigm. The exit competencies, outline, and course description reflect this design. The syllabus contains a full course description, detailed exit competencies, and a general course outline. The student clearly is told that the objective of our course is to learn both the object-oriented paradigm as well as specific object-oriented techniques. The tools to accomplish this process are Java as our programming language and UML as our notation and modeling tools.

Learning object-oriented applications is seen as primarily a three-step process of OO concepts, Analysis and methods, and Language specifics – in this case Java

**Clarity.** Every portion of the course requires clarity. The detailed syllabus noted clear assignments and clear instructions on timing and submissions. The entire syllabus presented unambiguous instructions. A detailed weekly schedule was provided for clarity on what is expected every week of the term. Since class meetings do not exist for online courses, the author has also recognized the need for structure. A detailed schedule provides both clarity and structure to the learning experience. With limited opportunity for interaction, specific and detailed instructions are provided on all aspects of the class including reading assignments, grading, assignment submission, deadlines, location and methods of submission, solution feedback, opportunity for resubmission (or lack thereof based on school policy), and primary and secondary means of communication.

**Feedback and Communications.** In developing programming solutions there are inevitable syntax and construct issues that arise. To maintain motivation and enthusiasm in students it is necessary to respond to students on a frequent and timely basis. My commitment is to respond to student questions within 24 hours. For emergency situations, I encourage my students to phone me at any time. Many students have taken advantage of this. Sometimes asynchronous communication is not adequate. I recently had a student who did not understand the submission or assignment concepts. A cell phone call from him solved all difficulties. I have recently also added the opportunity for synchronous chats via AOL Instant Messenger. This has proven useful as well. Assignment grading is performed within 4 days of final due date. With a somewhat compressed schedule, it is imperative to provide detailed feedback prior to the start of the next assignment.

**Rapport.** The development of a positive relationship with distance students is started even before the class begins. At least one month prior, the course syllabus is posted on the course website. The author's email and cell phone number are prominently displayed on the site and communication is encouraged prior to the class to handle any concerns or questions. Students and potential students frequently take advantage of this option, frequently asking about course assignments, texts, or prerequisites. The development of rapport continues throughout the term with frequent email communications, online forum discussions, and cell phone communications.

**Technical support.** One of the essential ingredients of online success is to have a stable and effective technical infrastructure for communications, assignment submission, and feedback. If the technical environment causes problems or delays for the student, the entire educational experience is threatened. The school where this course was taught had a tested home developed software environment for forum activity, assignments, and electronic mail. The school has now transitioned to another excellent environment, WebCT for all online course activities. The hardware infrastructure, including both server and telecommunications, is also vital.

**Variety.** One of the essential ingredients in my experience with online courses has been to provide variety in pedagogical methods to allow for different type of learning. Though the course by its very nature requires development and practice of actual programming, there are a variety of methods to cultivate an understanding of both the theory and practice. The variety of processes were specifically detailed in my OO application syllabus and included Text Reading, Lecture Notes, Forum Discussion, Analytical Assignments, Programming Assignments, and a comprehensive three-tier project. The course supports multiple forms of visual learning (with text readings and lecture notes), discussion learning (through forums) and analytical learning (through UML modeling). This is in addition to the actual programming assignments.

**Hands-on experience.** Hands-on learning is accomplished through the programming assignments themselves, the UML modeling and the completion of a capstone comprehensive three-tier application project. Another major aspect of the hands-on learning experience is the actual choice of specific programming tools to accomplish the project. Students were encouraged to use the programming tool of their choice. One specific assignment called for the use of the Forte IDE which was bundled with the text. This was a required hands-on activity. The students were required to gain experience with at least one IDE. Though some students complained about the age of the IDE, the use of older technology is common in the real world and constituted a valuable flexibility lesson. An interesting variety of development environments were used by the students in the 2004 class. For those who specified their choice, the following were used Eclipse – 4, Ready to Program – 4, Java 2 SDK -7, JBuilder – 4, JCreator – 1, JGrasp – 4, and NetBeans - 2.

The specific programming assignments are also included: UML exercises, application logic assignments, interface layer tasks, and data access classes. There was a major emphasis on hands-on activities, first with actual development of UML diagrams and then various programming assignments dealing with an aspect of the three-tier hierarchy. The final capstone project was a full three-tier development solution. The culmination of the coursework required an understanding of a three-tier implementation as well as knowledge, creativity, and

complexity. UML requirements included class diagrams, use case diagrams, and appropriate sequence diagrams.

**Content.** The content of the course for OO programming was shown to be an especially important aspect for success. Previous sections have detailed the content in terms of assignments and deliverables. The full content of the course is detailed in the variety and content of the course. One of the additional major requirements of teaching OO is the selection of textbooks. Without the ability to demonstrate techniques live, the texts must be able to guide the students through the material with simple easy to understand examples. The texts which I have found which accomplish this are *The Object-Oriented Approach: Concepts, System Development, and Modeling with UML* by John w. Satzinger and Tore U. Orvik, 2001. ISBN 0-619-03390-8 and *Object-Oriented Application Development Using Java*, by E. Reed Doke, John W. Satzinger, and Susan Rebstock Williams, Course Technology, 2002. ISBN 0-619-03565-X. The textbooks were viewed favorably. A typical student comment was “I also thought the book was a wise choice. Whenever I got stuck, I was always able to refer to the book easily and quickly. A major supplement to the textbook are links to numerous Internet sources. This collection which I have collected over several years provides valuable clarification to troublesome issues.

## CONCLUSION

Teaching object-oriented applications via distance education presents the same challenges that all distance education courses face. There is a need for fast and clear feedback on all assignments. The establishment of a strong rapport with students is difficult given the distance and time barriers. But I have found that OO applications require even more attention to course content than other courses. There must be a very clear and strong textbook. Assignments are structured very carefully to build on past knowledge but not overwhelm the learning experience. Exposure to IDE tools are a major component of the course since productivity of the programming effort is a vital issue in the field today. Finally, the concept of a overall capstone experience is carefully constructed to include all aspects of OO analysis, design, and implementation. The challenge of teaching OO applications via the Internet is not insurmountable. By heeding the key success factors, the OO experience can generate these actual comments - “it was the most useful and tied together problems that I experienced with other classes. It was better than all the Java classes offered and the Software (OOA & OOD) put together.” In the future, a comprehensive quantitative assessment plan will be pursued to quantify the qualitative findings of this study.

## REFERENCES

1. Abraham, S. (2002) Teaching a Programming Class Online: A Faculty Perspective, In D Colton, M J Payne, N Bhatnagar, & C R Woratschek (Eds.), *The Proceedings of ISECON 2002*, (19) (San Antonio): 223c. AITP Foundation for Information Technology Education.
2. Alley, L. & Jansak, K. (2001). The ten keys to quality assurance and assessment in online learning, *Journal of Interactive Instruction Development*, 3-18.
3. Burton, P. & Bruhn, R. (2003). Using UML to Facilitate the Teaching of Object-Oriented Systems Analysis and Design, *Journal of Computing in Small Colleges*, 19(3), 278-290.

4. Eastmond, D. (2000). Enabling student accomplishment online: An overview of factors for success in web-based distance education, *Journal of Educational Computing Research*, 23(4), 343-358.
5. Hara, N. & Kling, R. (2000). Students' distress with a web-based distance education course: An ethnographic study of participants' experiences, *Center for Social Informatics*. Available at <http://www.slis.indiana.edu/CSI/WP/wp00-01B.html>
6. Hillesheim, G. (1998). Distance learning: barriers and strengths for students and faculty, *The Internet and Higher Education*, 1(1), 1998, 31-44.
7. Jegede, O., Taplin, M., Fan, R., Chan, M., & Yum, J. (1999). Differences between low and high achieving distance learners in locus of control and metacognition, *Distance Education*, 20(2), 255-273.
8. McGill, T., Volet, S., & Hobbs, V. (1997). Studying computer programming externally: who succeeds? *Distance Education*, 18(2), 236-256.
9. Molstad, L. (2001). Teaching Computer Programming Using Distance Education Technology, *Journal of Computing in Small Colleges*, 17(1), 265-277.
10. Piercy, K. (2000). Teaching gerontology via distance education: variety is the key to success, *Educational Gerontology*, 26, 665-675.
11. Soong M.H.B., Chan, H., Chua, B., & Loh, K. (2001). Critical success factors for on-line course resources, *Computers and Society*, 36, 101-120.
12. Stidham, S. & Frieden, B. (2002). Ten easy steps to online success, *Business Education Forum*, 47-49.
13. Thomas, R. (2000). Experiences Teaching C++ Programming Online, *Journal of Computing in Small Colleges*, 15(5), 214-222.
14. U.S. Department of Education, National Center for Education Statistics. (2003). Distance Education at Degree-Granting Postsecondary Institutions: 2000–2001, NCES 2003-017, by Tiffany Waits and Laurie Lewis. Project Officer: Bernard Greene, Washington, DC Available at <http://nces.ed.gov/pubs2003/2003017.pdf> , 2003.
15. Wang, S. (1994). Basic considerations of distance education programs, *International Journal of Instructional Media*, 21(1), 53-60.