Techniques for Engaging Students in an Online Computer Programming Course

Eman M. EL-SHEIKH
Department of Computer Science, University of West Florida
Pensacola, FL 32514, USA

ABSTRACT

Many institutions of higher education are significantly expanding their online program and course offerings to deal with the rapidly increasing demand for flexible educational alternatives. One of the main challenges that faculty who teach online courses face is determining how to engage students in an online environment. Teaching computer programming effectively requires demonstration of programming techniques, examples, and environments, and interaction with the students, making online delivery even more challenging. This paper describes efforts to engage students in an online introductory programming course at our institution. The tools and methods used to promote student engagement in the course are described, in addition to the lessons learned from the design and delivery of the online course and opportunities for future work.

Keywords: Online learning, introductory computer programming, online course delivery, student engagement, computer science education.

1. INTRODUCTION

A rapidly increasing demand for flexible educational alternatives has led to a significant rise in online program and course offerings. Many higher education institutions across the country are increasingly developing online programs courses that were once only available in a face-to-face setting. This case is true at our university, where there is an increasing trend in the number of online courses and programs being offered. This increase in online course offerings is designed to meet the changing needs of our students. At our institution, 54% of students work more than 20 hours off-campus, and 62% are providing care for dependents living with them. Many institutions nationwide are reporting similar trends.

One of the main challenges that faculty who teach online courses face is determining how to effectively engage students in an online course format. Courses that are delivered online usually have the same objectives and student learning outcomes, and should aim to engage students in meaningful learning experiences similar to their face-to-face equivalents. Recent reports indicate that the skills that are most important to employers in hiring recent college graduates are teamwork skills (44%), critical thinking and reasoning skills (33%), oral and written communication skills (30%), and the ability to assemble or organize information (21%) [1, 2, 3]. The design and delivery of online courses should support the development of these essential skills. In addition, online computer programming courses should promote the development of effective programming and software development skills.

This paper describes efforts to engage students in an online introductory computer programming course. The course was developed using Desire2Learn™ [4], the course management system that has been adopted at the University of West Florida. The course described is a foundational Java programming course offered by the Department of Computer Science. It is the first course that students majoring in a computing discipline take. The course is also taken by students who chose to minor in one of the computing programs or students with a general interest in programming, making the course additionally challenging to teach.

The next section reviews current literature related to online programming courses. Section three describes the design and structure of the online course. The methods and tools that were identified and used for engaging students are described next. Finally, the paper concludes with a discussion of the lessons learned in the process and opportunities for future work.

2. REVIEW OF RELATED LITERATURE

A review of the literature related to the development and delivery of online courses reveals a number of important issues. Maki and Maki [5] conducted an extensive survey of online courses in various disciplines, including computer science, economics, biology, statistics, psychology, and education. Their results indicated that learning outcomes achieved by students in online courses were no worse than, and often no better than, those achieved by students in face-to-face courses. They concluded that the activities in which students participated in were more important than the medium of delivery for achieving the goal of improving student learning outcomes.

Some of the literature focuses specifically on the design of online programming courses, including reports of technologies and methods used in such courses, and of their impact on student learning outcomes. Fisher and von Gudenberg [6] discussed the presentation of a Java course online. Two major technologies formed the basis of the course: a hypermedia tutorial system that provided significant interactivity with examples and exercises, and a semi-automated assessment scheme that improved feedback time to the students. The course did not include synchronous activities, and, while the completing students produced highly satisfactory work in the opinion of the authors, attrition rates were approximately 50%.

Reeves, Baxter, and Jordan [7] described an introductory course in C++ taught online with the WebCT Learning Management System (LMS). The course was built around this LMS which contained the syllabus, daily assignments, longer-term programming projects, PowerPoint-based presentations, a textbook, and a programming environment. Students in the online class were encouraged to attend a face-to-face section of the same course (at least at the start of the term), but only one student took advantage of this opportunity. While the opportunity for synchronous communications was present, it was not available online and largely ignored.

Thomas [8] described an online offering of a C++ programming course. She related that even though the course was geared toward mature students with at least a year of programming experience, face-to-face meetings with a teaching assistant proved useful. Despite the fact that the course included utilization of standard distance communications, several students reported feeling isolated and wishing for better contact with the instructor. Suggestions included holding face-to-face exam reviews. Thomas concluded that online courses require mature and motivated students and some compensation for the lack of face-to-face interactions.

Zachery and Jensen [9] describe the organization of a course in JavaScript programming that was offered online. Their course included custom materials that they called "example-based narratives" which illustrated the results of executing pieces of code, coupled with a *hint* facility for exercises. Programming assignments required completion of skeleton code provided by faculty and did not include hints. Students could get help either via email or telephone calls to faculty during office hours, maintaining a synchronous aspect to the course.

Molstad [10] describes uses of various types of distance educational technology in an online introductory programming course, including the use of two-way audio-video capabilities that were used to allow students to access recordings of lectures. Students could ask questions that arose in the process of viewing the recordings, but no synchronous capability was available.

Varying results regarding the quality of outcomes that are achieved in online versus face-to-face programming courses exist in the literature. Ury [11] concluded that the performance of online students was satisfactory, but that their aggregate final grade was significantly lower than that of students who took an equivalent face-to-face class. Kleinman and Entin [12] arrived at a different conclusion, reporting that there were no significant differences in overall outcomes, corroborating the results of Maki and Maki [5].

Other results in the literature indicated the need to find ways to improve attrition rates in addition to learning outcomes. El-Sheikh, White, and Coffey [13] reported relatively little difference in outcomes between face-to-face and online students in an introductory programming course, but noted that dropout rates were much higher in the online section of the course. Reeves, Baxter, and Jordan [7]

corroborated this result. They reported that while completers of the online version of the course performed about as well as those from the face-to-face class, the online section had double the attrition rate.

Some articles in the literature address the issue of student engagement in online courses, although not specifically for programming courses. Lytle, Lytle, and Brophy [14] discussed methods to promote student engagement in relatively large online classes. They included activities that focused on early motivation and participation, active engagement with the content being covered, and use of a simulation to intensify participation and retention. Arbaugh [15] studied the impact of technologies, pedagogy, and student characteristics on student engagement in online Master's of Business Administration (MBA) courses. The results revealed that the most significant factors towards student learning were related to the instructor's efforts to create an interactive classroom environment.

This brief literature review reveals some of the important issues that are relevant to the development and delivery of online computer programming courses. A variety of technologies and methods have been employed and analyzed. Varying results exist regarding the level of achievement and attrition rates in online versus face-to-face classes. The design and incorporation of activities and tools that can foster better student engagement would seem to be of value in improving attrition rates and learning outcomes. The remainder of this paper will describe the development of an online introductory programming course, the tools and activities used to promote student engagement in the course, and the lessons learned from offering the course.

3. COURSE DESIGN AND STRUCTURE

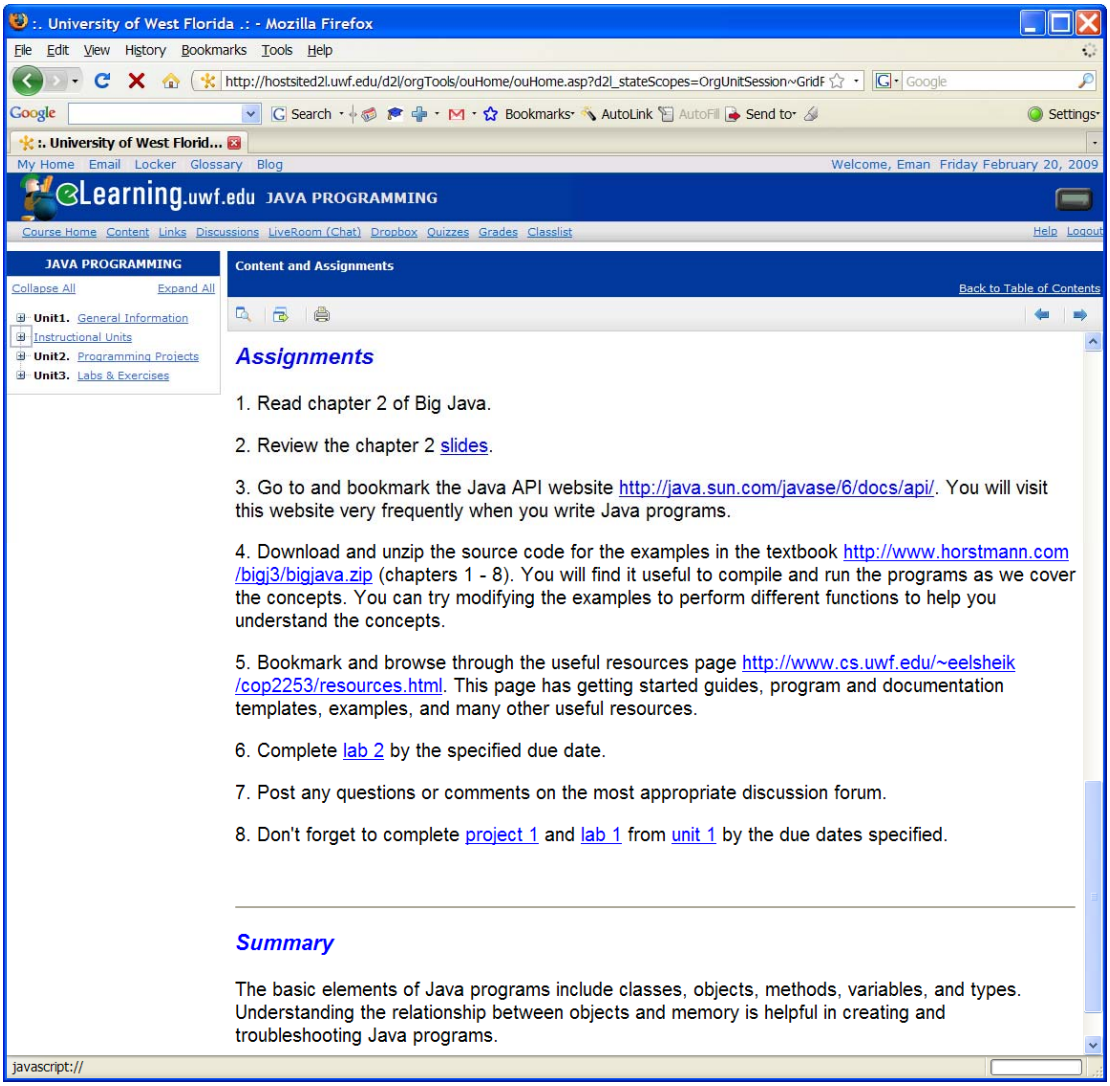
The online course was developed using Desire2Learn™ [4], the course management system adopted by our institution. This course management software provides an integrated framework for developing the components of an online course, as well as for delivering the course. Since the course was to be delivered completely online, it was important to develop a clear organization for the course materials that would provide a range of content, resources, and activities, and that would enable students to easily locate and access relevant materials. The course web site was organized into categories including:

- announcements and course calendar
- content, resources, and assignments
- discussion boards
- quizzes and exams
- drop boxes
- class list and grade book

The content was organized around eight instructional units. Each unit included a summary page with an overview of the material, and links to related assignments and resources. This allowed students to access a single page for each instructional unit to find everything that would be relevant to their learning goals at any time during the semester. In addition, a phased delivery system was used for the instructional units, in which each unit was made accessible to students at the time it was to be covered in the course schedule. This helped prevent students from feeling lost or overwhelmed with too much material by directing their attention to each unit as it was being covered. A screen shot from the course web site, which shows an overview of the instructional units and other materials, is shown in figure 1.

In addition to the instructional units, the course also incorporated a large set of resources, including programming examples, content slides, getting started guides, programming templates, and links to useful web sites, such as Java tutorials and textbook resources. Providing a diverse collection of resources was a design objective of the course to cater to students' various programming backgrounds and learning preferences. Through the course web site, students could also access the specifications and resources for all programming projects and assignments, and submit their projects and assignments using designated drop boxes. A screen shot illustrating the assignments section of an instructional unit is shown in figure 2.

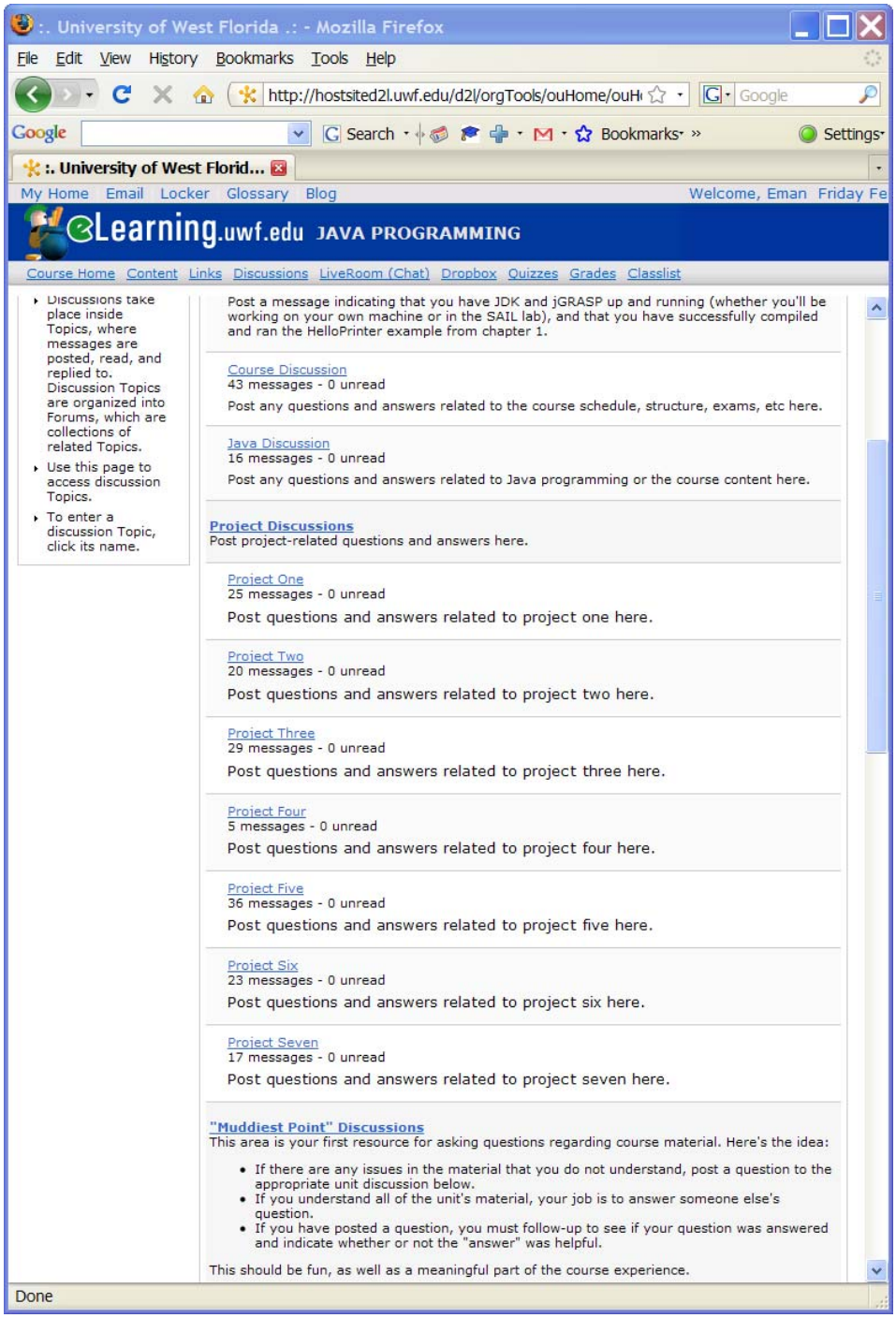
Fig, 1. Course web site: overview of instructional units and other materials



Fig, 2. Course web site: assignments section of an instructional unit

The course also included a set of discussion forums to allow interaction among the students and instructor about various topics and issues related to the course content and requirements. The discussion forums were organized into five categories: general discussions, lab-related discussions, project-related discussions, exam-related discussions, and "muddiest point" discussions, which were intended to provide a forum for students to discuss the issues in each instructional unit that were difficult to understand or that they had questions about. The variety of discussion forums allowed students to identify the most appropriate area to post questions and answers related to the course policies and schedule, assignments, content, or exams. Students were encouraged to post their questions or comments to the most appropriate discussion forum, as well as to try to answer questions posted by other students. A screen shot showing some of the discussion forums is shown in figure 3.

Chat rooms were also created as part of the course web site to allow synchronous interaction among the students and faculty to discuss course-related issues. These chat rooms provided a supplementary means of communication in addition to the asynchronous discussion forums. Students took two exams during the semester and a final exam at the end. The mid-semester exams were administered online. The final exam was proctored and common to both the online and face-to-face sections of the course. Students in remote locations were required to register at an approved testing center to take the final exam.



Fig, 3. Course web site: discussion forums

The course web site also provided a number of other useful features, including a page for announcements and reminders for upcoming deadlines, drop boxes for students to submit their work, and a section that enabled students to take online exams. This was used to administer the mid-semester exams in a way that gave students a 3-day window to take the exam to accommodate their schedules, but once they started the exam, it had to be completed within a certain time limit. The web site also included a grade book that displayed each student's grades in relation to other students' and the class average, a class list that allowed students to see which other students were currently logged into the course web

site, and a capability to email other students or the instructor individually or as a group.

4. METHODS AND TOOLS FOR PROMOTING STUDENT ENGAGAGEMENT

Several activities and tools were included in the online programming course to promote active learning and student engagement. The course incorporated two types of methods for promoting student engagement: asynchronous and synchronous methods. Asynchronous methods and tools, such as discussion forums and email communication, were used to provide opportunities for students to interact with other students and the instructor as their time allowed. These tools provide flexible interaction alternatives, and consequently, cater to the needs of students who choose to enroll in online courses and programs. Synchronous methods and tools, such as chat rooms and collaboration tools, were used to engage students in real-time active learning experiences. These activities helped transform the online course into a virtual active learning environment similar to a face-to-face classroom.

A wide variety of discussion forums provided the students and instructor with opportunities to engage in discussions about the course requirements, materials, and assignments. The discussion forums were widely used by students to post their questions, responses to other students' questions, resources that they wanted to share, and general comments about the course or Java programming. Students regularly responded to each other's questions and comments. The instructor monitored the discussion boards routinely, answering questions or pointing out useful resources when appropriate. The instructor also often posted leading questions, intended to initiate a discussion among students about a topic or issue relevant to the instructional unit currently being covered. The discussion forums helped create a *community of learners* in the course, with everyone contributing towards the achievement of common learning goals in various ways.

In addition to the asynchronous discussion forums, the course included the use of two synchronous tools and activities to foster greater student engagement. One of these activities was the incorporation of weekly synchronous class sessions using a collaboration software tool called Elluminate™ [16]. These weekly sessions, which were scheduled at the time most preferred by the majority of students, allowed the students and instructor to have virtual discussions about the course content and assignments and work together to complete related activities.

The collaboration software tool allowed participants to interact via voice, video, and text messages, and to access a common workspace that could be used to discuss course materials, assignments, or example programs. Another feature of Elluminate™ allowed participants to share their desktop, so that the instructor could develop, compile, and run programs while the students were observing or participating. In addition, students could give the instructor permission to take over their desktop to help answer questions or resolve errors in their programs.

The sessions were recorded and made available online so that all students could later view them. Figure 4 shows a screen shot from one of the synchronous sessions illustrating application sharing among the instructor and students. These synchronous sessions helped promote active learning and student engagement by providing additional opportunities for the students and instructor to continue to evolve as a community of learners and work towards fulfilling the course's learning objectives. Figure 5 shows another screen shot from one of the synchronous sessions illustrating a group activity related to one of the assignments.

Chat rooms were also created as part of the course web site to allow real-time interaction among the students and instructor. Several chat rooms were created for different topics, including a chat room for course assignments and requirements, and another one for discussions about object-oriented and Java programming techniques. Students could join these existing chat rooms or create their own chat rooms to discuss issues that emerge during the semester. The chat rooms were moderated by the instructor. They were available throughout the semester, and were often used to follow-up on questions that had been posted on the discussion board when there was a need for more direct, interactive discussion.

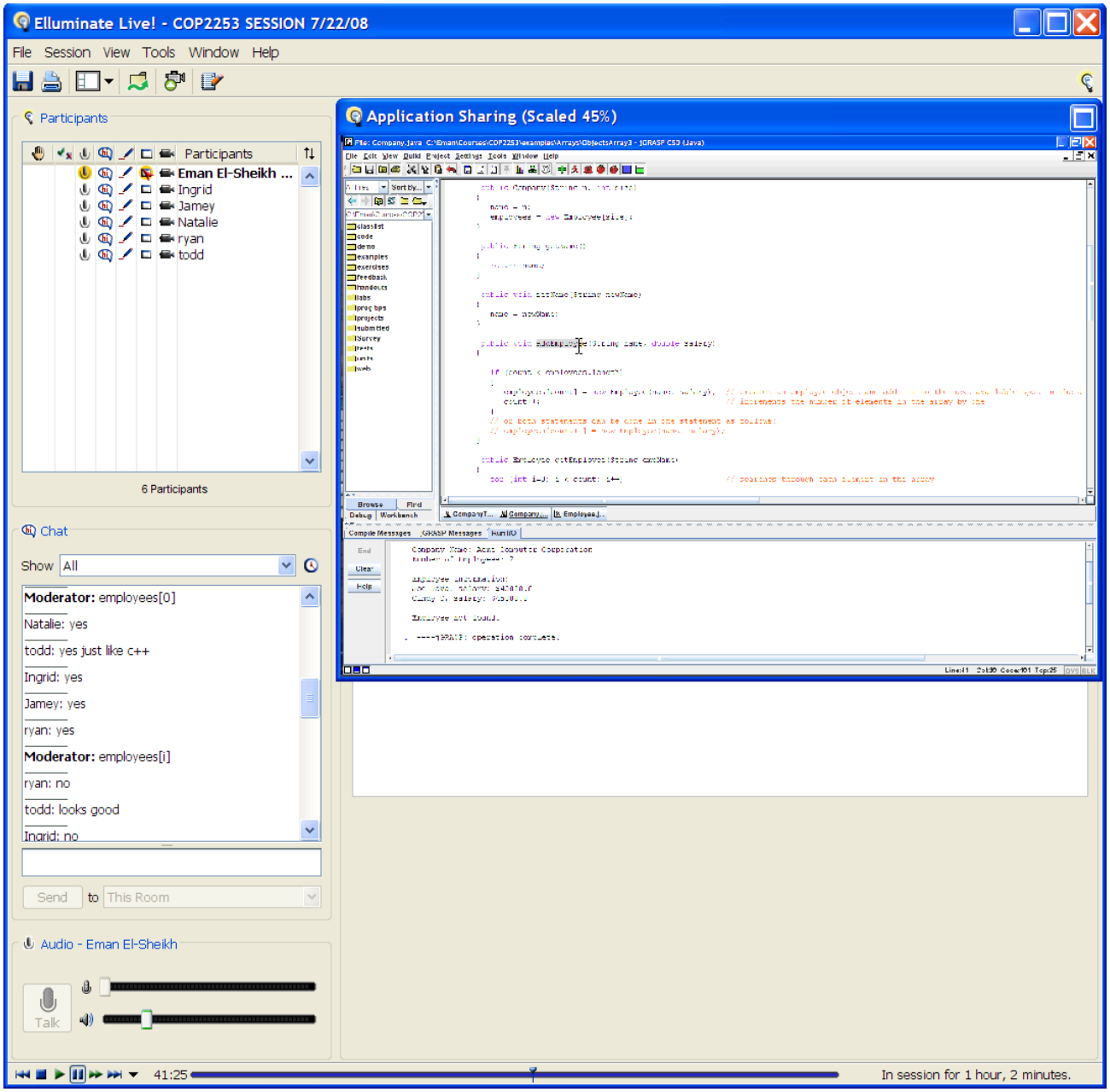


Fig. 4. Synchronous learning session: application sharing

Another activity integrated into the course design was the use of virtual office hours. Each week, the instructor had a dedicated time during which she was available online to allow students to get immediate assistance. During that time, the instructor was logged in to the course web site and could communicate synchronously with the students via the chat rooms, discussion boards, or a tool known as the pager. The pager is a feature of the course management system that allows students to "page" the instructor when they had a question, and the instructor could, in turn, respond immediately. The virtual office hours offered a regularly scheduled time during which students knew that they could get immediate assistance.

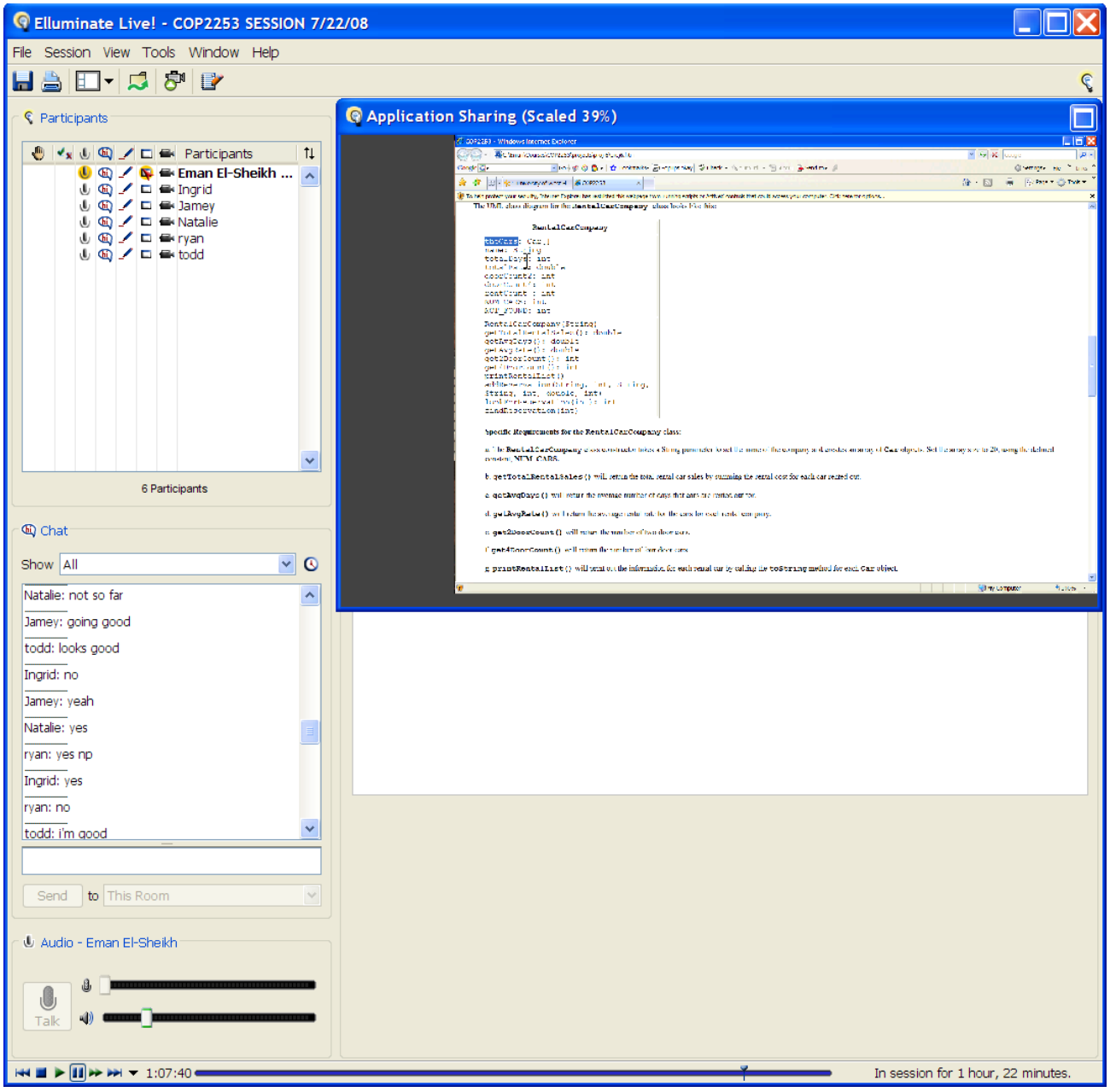


Fig. 5: Synchronous learning session: group assignment

Students could also get assistance using email, although this was discouraged except for questions that were very specific to an individual student's situation. Students were encouraged to use the discussion forums or chat rooms for general questions to emphasize the community of learners approach. A teaching assistant was available at the main campus during scheduled hours to help students who preferred to get face-to-face assistance, and also to provide online assistance via the discussion forums.

5. CONCLUSIONS AND LESSONS LEARNED

The programming course was delivered online using the methods and tools described in the previous section. Several lessons were learned from the design and delivery of the online programming course that can be helpful in offering future online programming courses, as well as in the design of other online computing-related courses. The major recommendations are summarized below:

- It is important to provide multiple methods for interaction to cater to a wide variety of students' preferences and learning needs. The course included numerous ways to facilitate interaction among the students, instructor, and teaching assistant, including discussion forums, chat rooms, virtual class sessions, virtual office hours, and the pager tool. Including a variety of methods to interact helps students stay engaged with the course, by providing multiple opportunities for them to have discussions about the course content and requirements, and get answers to their questions.
- Include opportunities for both synchronous and asynchronous interaction. Synchronous activities, like the virtual class sessions done using Elluminate™ [16], virtual office hours, and chat rooms allow students to get immediate responses to their questions. This is important for a programming course, in which students often have debugging, syntax, or 'how-to' questions that can interrupt their progress if not answered in a timely manner, yet can often be answered fairly quickly. Also include asynchronous interaction mechanisms, such as discussion forums, that enable students to seek assistance and engage in discussions as their time permits. Providing a variety of synchronous and asynchronous interaction methods further promotes student engagement in the course by catering to their range of learning needs.
- Identify appropriate technologies and tools that can help engage students in an online course. A variety of tools exist that can be used to enhance online learning experiences, and these tools are rapidly evolving. It is important to keep up with the availability of such tools and explore how they can be utilized to create active learning opportunities that can enrich the online learning environment. For example, the ability to demonstrate the design, implementation, and testing of programs greatly enhances the teaching of programming skills. The use of the Elluminate™ software tool enabled the instructor to demonstrate the development of programs, as well as to view and debug students' programs. In addition, the software facilitated collaborative activities and real-time discussions and interaction. Discussion boards and blog tools can facilitate discussions and interaction that keep students engaged throughout the semester. A pager or instant messaging tool can be used to let the instructor know that a student requires immediate assistance.
- Incorporate activities that enable collaboration to give students additional opportunities to interact with each another and create a community of learners. For example, the synchronous sessions included in the course can be used to facilitate collaborative learning activities in which students are divided into small groups of 2 – 4 students to solve programming problems. The sub-groups can then reconvene and discuss their results with the entire class. As another example, the chat rooms can be used to facilitate weekly group discussions.
- Another important requisite for student engagement in an online course is to require regular interaction and graded work as part of the course requirements. Students often enroll in online courses instead of face-to-face courses because they have work and/or family obligations, and are looking for flexible learning opportunities. With other demands on their time, students can easily get disengaged from an online course, which often adversely impacts their chances of succeeding in the course. Requiring regular graded assignments and related interaction opportunities helps students stay engaged throughout the semester. In this course, students were required to submit weekly programming assignments, with a set of review exercises due each week and a larger programming project due every other week.
- Requiring regular graded work and interaction will be more beneficial if students' progress is monitored regularly. It is important to track students' participation in the course, and find ways to engage students who are having difficulties keeping up with the material or whose performance is deteriorating. Indicators of students' participation in the course can include submission of assignments, contributions to discussion forums, and participation in the virtual class sessions.
- Online students benefit from more regular and detailed feedback on how they are doing in the course, compared to face-to-face students, who can often get informal feedback or indicators of their performance from the class sessions.

Ideally, students should receive detailed and timely feedback on all graded assignments and exams, preferably within a week after students submitted them. Providing students with more informal or brief feedback about their participation in course activities, such as the synchronous learning sessions, helps students stay engaged in the course. Such feedback also helps students reflect on their learning experiences, which can consequently lead to better achievement of the intended learning outcomes.

- In addition to providing useful feedback to the students, it is also important to ask for students' feedback during the semester and make changes to accommodate their needs as necessary. Online students often have varied learning needs and preferences, compared to face-to-face students. Administering a background survey at the beginning of the semester can help the instructor tailor the course materials and assignments to the students' specific learning needs. Asking for their input during and at the end of the semester on how well they achieved the learning goals and how to improve the learning experience can provide the instructor with useful suggestions for improving learning experiences and increasing student engagement for the remainder of the course or for future course offerings.

The lessons learned have generated several useful ideas for the design and delivery of future online programming courses. One technique that has shown potential in improving learning experiences and engaging students in online programming courses is the integration of synchronous learning sessions into the course. More studies are needed to determine the best ways of incorporating such synchronous activities for improving students' learning outcomes. El-Sheikh, Coffey, and White [17] investigated the use of synchronous learning sessions in an online programming course, and reported the difference in outcomes between face-to-face and online students.

Another promising path for future work is the investigation of strategies for incorporating collaborative learning opportunities in an online course environment. Identifying appropriate tools for facilitating collaborative activities and group interaction can help improve student learning outcomes in an online programming course. Further work is needed to design and integrate effective collaborative learning activities into the course. In addition to increasing student engagement, these strategies can promote the creation and sustainability of a community of learners in pursuit of the course's learning objectives.

6. REFERENCES

[1] Chickering, A., & Gamson, Z. (1987). Seven Principles of Good Practice in Undergraduate Education. AAHE Bulletin, Vol. 39, pp. 3-7.

[2] National Leadership Council for Liberal Education and America's Promise (2007). College Learning for the New Global Century. American Association of American Colleges and Universities.

[3] Pascarella, E., & Terenzini, P. (2005). How College Affects Students (Vol. 2): A Third Decade of Research. San Francisco: Jossey-Bass.

[4] Desire2Learn™ (2009). Desire2Learn™: Innovative Learning Technology. http://www.desire2learn.com/. Accessed: February 15, 2009.

[5] Maki, R. H., & Maki, W. S. (2007). Online courses. In F. T. Durso, S. Dumais, S. Lewandowsky, and T. J. Perfect, Handbook of Applied Cognition, Second Edition (pp. 527-552). West Sussex, England: John Wiley and Sons.

[6] Fisher, G., & von Gudenberg, J. W. (2006). Improving the quality of programming education by online assessment. Proceedings of Principles and Practices of Programming in Java, PPPJ'06. Manheim, GA. ACM Press. pp 208-211.

[7] Reeves, T., Baxter, P., & Jordan, C. (2002). Teaching Computing Courses - Computer Literacy, Business Microcomputer Applications, and Introduction to Programming Online Utilizing WebCT. Proceedings of the 11th Annual Rocky Mountain Conference of the Consortium for Computing Sciences in Colleges, Farmington, NM. pp. 290-300.

[8] Thomas, R. (2000). Experiences Teaching C++ Online. Journal of Computing Sciences in Colleges, Vol. 15, No. 5. pp. 214-222.

[9] Zachery, J.L., & P.A. Jensen, (2005). Exploiting Value-Added Content in Online Course: Introducing Programming Concepts with HTML and JavaScript. Proceedings of SIGCSE 2005: The 36th ACM Technical Symposium on Computer Science Education, Reno, NV. ACM Press, New York, pp. 396-400.

[10] Molstad, L. (2001). Teaching Computer Programming Using Distance Education Technology. Journal of Computing in Small Colleges, Vol. 17, No. 1, pp. 265-277.

[11] Ury, G. (2004). A Comparison of Undergraduate Student Performance in Online and Traditional Courses. Journal of Consortium for Computing Sciences in Colleges, Vol. 19, No. 4.

[12] Kleinman, J., & Entin, E. (2002). Comparison of In-Class and Distance-Learning Student Performance and Attitudes in Introductory Computer Science Courses. Journal of Consortium for Computing Sciences in Colleges, Vol. 17, No. 6.

[13] El-Sheikh, E. M., White, L. J., & Coffey, J. W. (2007). Reflections in Uncharted Waters: Teaching Foundational Programming Online. Proceedings of EISTA 2007: The Fifth International Conference on Education and Information Systems, Technologies and Applications. Orlando, FL July 12-15, 2007.

[14] Lytle, S., Lytle, B. & Brophy, J. (2006). Student Engagement in Large Undergraduate Online Courses. In T. Reeves & S. Yamashita (Eds.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2006 (pp. 1330-1333). Chesapeake, VA: AACE.

[15] Arbaugh, J. B. (2000). How Classroom Environment and Student Engagement Affect Learning in Internet-based MBA Courses. Business Communication Quarterly, Vol. 63, No. 4, pp. 9-26.

[16] Elluminate™ (2009). Elluminate™: eLearning and Collaboration Solutions Software. http://www.elluminate.com/. Accessed: February 15, 2009.

[17] El-Sheikh, E., Coffey, J., and White, L. (2008). Exploring Technologies, Materials, and Methods for an Online Foundational Programming Course. International Journal of Informatics in Education, vol. 7, no. 2, pp. 1-18.