Informatics in Education, 2008, Vol. 7, No. 2, 259–276

Exploring Technologies, Materials, and Methods for an Online Foundational Programming Course

Eman M. EL-SHEIKH, John W. COFFEY, Laura J. WHITE

Department of Computer Science, University of West Florida
Pensacola, Florida, USA
e-mail: eelsheikh@uwf.edu, jcoffey@uwf.edu, lwhite@uwf.edu

Received: September 2008

Abstract. Introductory computer programming courses are inherently challenging for a variety of reasons. With increased demands for online delivery, the use of effective technologies, materials, and methods that best support online learning is essential to maximize student success. This article describes a recent study conducted at our institution with an overall objective to improve the design and online delivery of a foundational course in Java programming. The online course included a variety of technologies and materials intended to improve student learning outcomes, including an online synchronous interaction component similar to teleconferencing. A comparison of students' backgrounds, perspectives, and outcomes in an online section of the course compared to a benchmark face-to-face section was conducted using various evaluation methods. The results indicated that online synchronous sessions and several other aspects of the course were beneficial toward improving online learning. Results of the study, conclusions, and other issues warranting further consideration are described.

Keywords: online course delivery, teaching introductory programming, online versus face-to-face learning, virtual synchronous activities, course technologies.

1. Introduction

Introductory computer programming courses typically pose a challenge due to the variation in students' background and experience, the manner in which the material builds on itself, and the extensive time required to complete programming projects. They are usually important courses for students majoring in computing, information technology, or software engineering disciplines because they serve as the cornerstone of the curriculum. Despite best efforts by instructors, introductory programming courses typically evidence high attrition rates. Within this context of historic problems with student success in traditional programming classes, many instructors are being asked to offer these courses online. It is incumbent upon faculty to explore alternatives and adopt strategies that will maximize the likelihood of student success and retention under new course delivery modes.

This article describes a research study that explored a variety of technologies, materials, and methods for the design and delivery of an online introductory Java programming course. The overall goal was to determine how to improve online delivery of foundational programming courses, with the objective of improving student engagement and learning outcomes. It builds upon experience gained in a first online offering of the course and a prior study regarding that course. The first version of the course was comprised of self-study instructional units, numerous programming assignments, topic-specific discussion threads, online examinations, and other pedagogical aids. The current study added virtual synchronous learning activities to the online course via an academically-oriented collaboration software package named Elluminate, and yielded significant new data derived from students' responses to questionnaires, surveys administered at the beginning, mid-point, and end of the semester, and students' performance in the course. A face-to-face class taught in parallel by the same instructor provided a control group for this study.

The remainder of this work contains a brief description of literature that is relevant to teaching and learning programming, and online computing-related courses. Following this review, the design and organization of the online course are described in Section 3. This course was built upon two major software capabilities: a Learning Management System (LMS) and a Synchronous Groupwork System (SGS). These capabilities and how they were employed will be discussed. Section 4 identifies the research questions for this study, the methods, and the results. This article concludes with a summary and discussion of the results and lessons learned.

2. Literature Review

This literature review will address issues related to developing and delivering online programming courses including the range of technology and methods used, degree of support for interactivity or synchronous communications, and reported outcomes. Bowers (2007) describes the use of meeting groupware to introduce collaborative group work into an online Information Technology program. Recorded lectures were augmented with synchronous distance project work by students. In the first version of the course, all student activities were in a main virtual room. In a second version, groups worked in separate rooms before returning to report results in the main room. The authors found value in the group work, assessing it as leading to outcomes similar to face-to-face group work.

Fisher and von Gudenberg (2006) discuss 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, in the opinion of the authors, the completing students produced highly satisfactory work, attrition rates were approximately 50%.

Reeves *et al.* (2002) describe an introductory course in C++ taught online with the WebCT Learning Management System. The course was built around this LMS which contained the syllabus, daily assignments, longer-term programming projects, PowerPoint 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 largely ignored.

Zachery and Jensen (2003) 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 (2001) 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.

Thomas (2000) describes an online C++ programming course. She relates 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 concludes that online courses require mature and motivated students and some compensation for the lack of face-to-face interactions.

The literature contains conflicting findings regarding the quality of outcomes that are achieved in online versus face-to-face programming courses. Ury (2004) states that, in an absolute sense, 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 (2002) arrive at a different conclusion, reporting that there were no significant differences in overall outcomes. El-Sheikh *et al.* (2007) report relatively little difference in the outcomes of matched face-to-face and online Java courses, but note that dropout rates were much higher in the online version of the course. Reeves *et al.* (2002) report 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, a result corroborated by El-Sheikh *et al.*

This brief literature review reveals some of the major issues that are relevant to the effective design and delivery of online introductory computer programming courses. A variety of technologies have been employed, and attempts to provide capabilities that mimic those available in class are evident. In online courses, synchronous communications are supported through a range of technologies including telephone calls, instant messaging, and meeting software. The range of materials and methods is as broad as those for face-to-face classes.

The question of the level of achievement in online versus face-to-face classes does not seem completely resolved with regard to introductory programming. An ability to foster efficient, real-time communications would seem to be of value. The next sections of this paper will provide a description of the second offering of an online course in Java programming in which synchronous communication capabilities were added. Results of a study conducted to assess the course structure and outcomes are presented.

3. Course Design and Organization

This section contains a description of the overall design and motivation for the course. The course structure is described, including the common course elements, course web site and usage, and the synchronous learning sessions conducted.

3.1. *Overview and Design Rationale*

The study was conducted in a course named “Java Programming”, an introductory-level programming course offered by the Department of Computer Science at the University of West Florida (UWF). It is the first of a three-course sequence of programming foundations for computing-related majors. The introductory programming course is also taken by students pursuing a minor in computer science, several other majors, and students with a general interest in programming. The combination of majors and non-majors enrolled in the course makes it more challenging to teach and more interesting to study. The work described here was conducted in two sections of the course offered in the fall 2007 semester. One section was delivered face-to-face with a 3-hour class meeting held once a week. The other section was offered online, via an online course management system named Desire2Learn, with a synchronous component delivered using Elluminate. The initial enrollment of the face-to-face section was 22 students, with a final enrollment of 19, while the online section started with 29, and had a final enrollment of 16. Although some components were unique to each section, several course elements were common to both sections, and are described next.

3.2. *Elements Common to the Face-to-Face and Online Versions of the Course*

Both sections of the course had the same instructor and the same goals, sometimes using different methods to achieve those goals as appropriate for online or face-to-face learning environments. A common set of learning outcomes applied to both sections. Students in both sections were assigned the same programming projects, had the same deadlines, were graded using the same criteria, and had access to the same teaching assistant (TA). The final exam, which was proctored, was also common for both sections. Students in both sections also took two exams during the semester, which covered the same material but were administered in different ways. The face-to-face section took their mid-semester exams in class, while the online mid-semester exams were administered using the course management system, Desire2Learn.

3.3. *Course Website Structure and Usage*

A dedicated website was developed for the course using Desire2Learn. Students in both sections had access to the course website. In order to enable students to find specific materials, the website was organized into categories including announcements, content, discussion boards, drop boxes, and a grade book. The content was divided into eight instructional units, each of which included a summary page with an overview of the material, and links to related assignments and resources. Having a single page for each unit that provided an overview of the content along with links to slides, examples, assignments, and other resources related to that content made it much easier for students to find everything relevant to each topic. This arrangement was motivated by student feedback from the previous semester's online course. Another design feature of the course was the phased delivery of the instructional units. Each unit was published just prior to its date on the schedule to avoid overwhelming students with too much information from the beginning of the semester or with resources that were not relevant to their current learning goals.

Another important component of the course website was the well-organized and numerous discussion forums. The website included five categories of discussion: general discussions, which were used for introductions as well as discussions regarding the course or Java programming in general; project discussions, which included a forum for each assigned project; lab discussions, which included a topic area for each assigned lab; "muddiest point" discussions, which included a forum for each instructional unit; and exams, which included a discussion topic area for each exam. The variety of discussion forums allowed students to identify the most appropriate area to post questions and answers related to course policies and schedule, assignments, content, or issues pertaining to an exam. The discussion boards became an instrumental tool for engaging students and facilitating interaction among them.

The course website also included an extensive set of resources for the students. These included course-specific resources, such as "getting started" guides, program and documentation templates, additional programming examples, content slides, links to self-review exercises, and exam review guides. The resources also included links to Java programming resources, integrated development environments (IDEs), programming guides, and textbook-related resources. The course website 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, a grade book that displayed each student's grades in relation to the class average, a class list that allowed students to see which other students were currently logged into the course website, and a capability to email the entire class, individual students, TA, or instructor.

3.4. *The Conduct of Synchronous Sessions*

One of the novel components of the online course was the synchronous sessions conducted using Elluminate, a software product that enables synchronous communication and interaction. The original design objective was to include a weekly collaborative session in which the students would initially work in sub-groups to collaborate on solving a programming problem, then reconvene to discuss the problem and solution all together. However, the actual use of the synchronous sessions evolved into a different format for a variety of reasons. Although the instructor encouraged participation by scheduling the sessions at the time most preferred by the majority of students, the number of students who participated in the synchronous sessions was not sufficient for effective subgroup work. However, the students did work collectively on the planned activities.

The reason for low participation is thought to be the optional nature of these sessions. Online courses at our institution are not scheduled with any corresponding time-blocks that permit instructors to require participation in synchronous activities. This lack of required sessions clearly impacted student participation. The weekly sessions were used to discuss the programming projects and lab assignments, and to discuss activities, examples, and questions relevant to each assignment. The synchronous communication software allowed students to interact using text messages, voice, and video. Elluminate also provides an application sharing feature that allowed the instructor to demonstrate the development, debugging, and execution of Java programs to students. The instructor could also take control of an individual student's desktop, which was useful to resolve a specific question, or to assist in finding program faults. The sessions were recorded and made available online so that all students could later view them.

Initially, the synchronous sessions were only available for students enrolled in the online section, but were later opened up to students in the face-to-face section to encourage more collaboration. Since the course involved so many different activities and technologies, it was deemed important to try to determine those that contributed to successful outcomes. A study designed to help make those determinations is described in the next section.

4. An Empirical Study in the Efficacy of the Online Course

The goal of this study was to determine what was most effective in the design of an online foundational programming course and how to improve online delivery of the course. An additional objective was to improve student engagement and learning outcomes. In this section, we describe our research questions, the methods used in our study, and summarize the results.

4.1. *Research Questions*

The following two research questions guided the study:

Question 1: What mix of technologies, materials, and methods is most beneficial in the design and delivery of an online foundational computer programming course?

Question 2: Will this online course yield learning outcomes that are comparable to a well-established face-to-face benchmark course?

A variety of data were collected to answer the research questions. Background information was collected on all students to create context for both questions. Data regarding participation in synchronous online learning sessions, pre- and post-course surveys administered by the investigators, the standard university course evaluation forms, and a separate survey from the university's Academic Technology Center provided inputs for question 1. Comparison of final averages between the groups provided quantitative data to address question 2. Each category of data and the results are discussed in the following sections.

4.2. *Methods*

Several methods were used to collect student input on the course. All students were given background questionnaires at the beginning of the semester with open-ended questions to identify their prior programming experience. Brief surveys were administered at the start and at the end of the semester that the study spanned. The surveys included basic demographic questions, questions on students' prior experience in online courses, as well as questions that asked students to rate the usefulness of online course components, and their interest and confidence in their programming skills. Additionally, our institution's Academic Technology Center (ATC), a unit that provides faculty support related to instructional technology, posted their own surveys on the course website for the students to complete at the mid-term and end of the semester.

The course included 14 one-hour long synchronous sessions, approximately one per week. As described in Subsection 3.4, the sessions could not be required, which impacted the number of students who participated. Each session focused on an assignment relevant to the material covered during that week. For example, for the unit on an elementary data storage structure named an array, the session focused on the assignment and activities related to arrays. During the sessions, the instructor discussed the assignment or activities with the students, asked relevant questions, and also responded to students' questions. In some cases, students also informally interacted with each other during the session.

As part of the state-mandated university requirements, students completed course evaluations at the end of the semester, and those results were analyzed as part of the study. Learning outcomes for face-to-face and online students were compared on common course elements, namely the programming projects and final exam. The next sections summarize student background, participation in the synchronous sessions, survey results, and learning outcomes.

4.3. *Student Background Information*

Questionnaires were administered to all students at the beginning of the course. The questionnaires indicated that the students' backgrounds were extremely varied, ranging from minimal computer usage to extensive experience involving multiple environments and applications. Information related to background experience with operating systems and software applications submitted in questionnaires by 21 face-to-face students and 16 online students are summarized in Figs. 1 and 2. The results indicate that the computer experience of students in both the face-to-face and the online sections was similar.

Students' experience with programming was also quite varied. Some students had no prior programming experience while others already knew several languages. Results regarding prior experience with programming in Java and other programming languages are

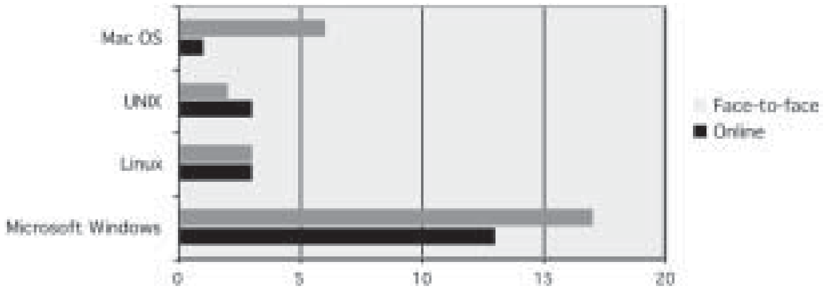

Fig. 1. Questionnaire results for students' prior experience with operating systems.

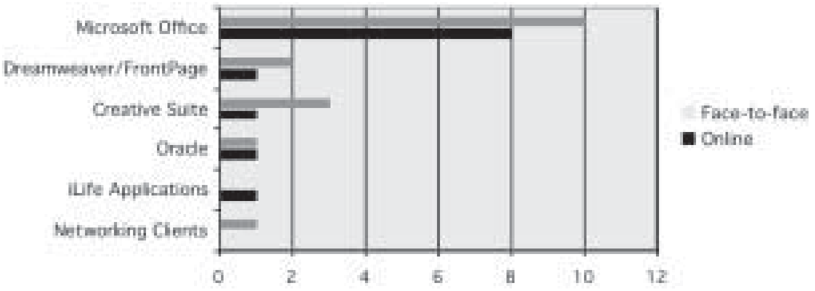

Fig. 2. Questionnaire results for students' prior experience with software applications.

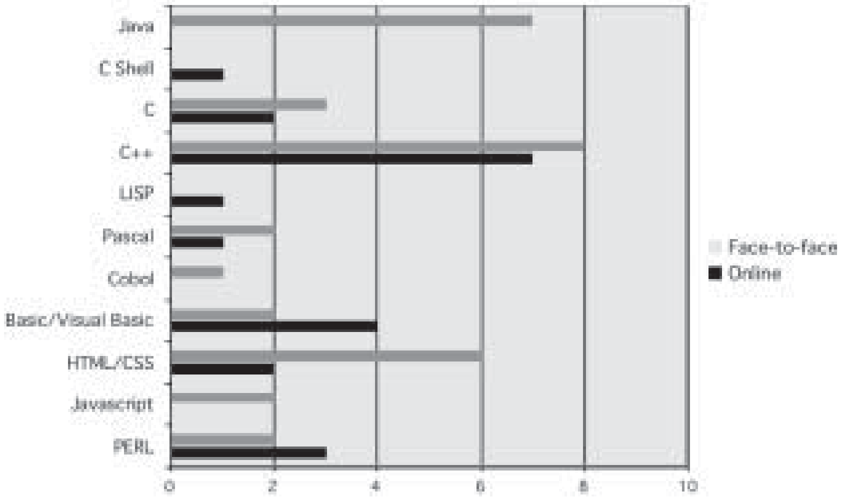

Fig. 3. Questionnaire results for students' prior experience with programming languages.

shown in Fig. 3. It is interesting to note that no students in the online section had any prior experience with Java programming. However, a third of the students in the face-to-face section had prior experience with Java. It is likely that this difference in background impacted learning outcomes and attrition rates between the face-to-face and online sections. Prior programming experiences in languages other than Java were comparable.

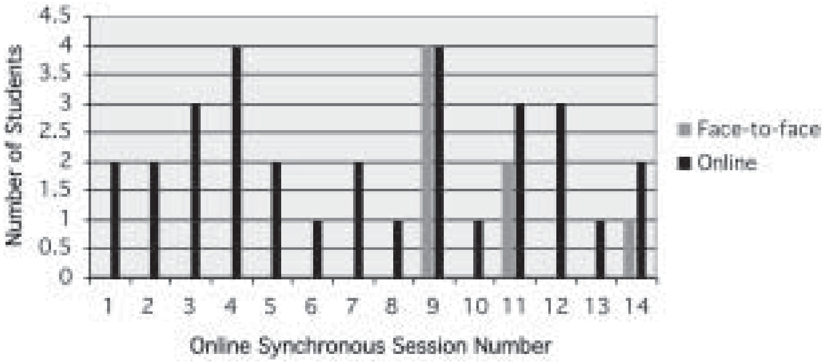

Fig. 4. Number of participants in the synchronous sessions.

4.4. *Participation in the Synchronous Sessions*

The number of students who participated in each synchronous session was recorded. Data regarding the number of times each recording was accessed would have provided interesting information, but was unavailable. Fig. 4 shows the total number of students who participated in each session, grouped by section. Although student participation in the online synchronous sessions was lower than expected, the data helps to shed some light on the usefulness of such sessions. The number of online students who participated in the sessions remained consistent throughout the semester, with an average of 2.3 students in the first half of the semester, and 2.1 in the second half of the semester. Sessions 1–8 were only available for online students, but the remaining sessions (sessions 9–14) were available for all students. The sessions were opened up to the face-to-face students in the hope of increasing participation and enabling more collaborative activities during each session. Despite this policy change, few face-to-face students (an average of 1.2) participated in the sessions.

Despite the overall low number of students who participated, it seems that many more students derived benefit from the sessions by later accessing the recordings. Although actual usage data on the number of students who later accessed each recording was unavailable, informal feedback received from students during the semester suggested that many who didn’t participate in the sessions routinely accessed the recordings later, and asked questions about the activities discussed in the recordings.

4.5. *Results of the Pre- and Post-Course Surveys*

Two surveys were administered to students in both sections, one at the beginning and one at the end of the semester. The first survey was taken by 13 students (6 female and 7 male), 8 from the online section and 5 from the face-to-face section. All but three students had prior programming experience and all but one had previously taken an online course. When asked about resources that helped most in prior programming courses, the most common student responses were textbook, programming assignments, and discussion threads. Table 1 summarizes demographics, and the results for students’ interest in

Table 1

Selected results from the initial survey

Category	Average Age	Gender	Interest in Programming	Confidence in Programming Ability
Face-to-face	33.4	2 Male, 3 Female	2.00	2.40
Online	34.4	5 Male, 3 Female	1.88	2.25
Total	34	7 Male, 6 Female	1.92	2.31

Table 2

Selected results from the final survey

Category	Average Age	Gender	Interest in Programming	Confidence in Programming Ability
Face-to-face	24.4	6 Male, 4 Female	2.50	2.60
Online	31.8	2 Male, 3 Female	1.80	2.80
Total	26.9	8 Male, 7 Female	2.26	2.73

programming and confidence in their programming skills. The two groups were similar with regard to age, interest, and confidence. The last two elements in Table 1 are mean scores on a 5-point Likert scale, where a response of 1 indicates very high and a response of 5 indicates very low. These results indicate a fairly high interest in programming with somewhat less confidence in initial abilities to program, in both the face-to-face and online classes.

The survey at the end of the semester was completed by 15 students. Table 2 provides a summary of the results of that survey. Interestingly, the students were younger and, as completers of the course, had survived the winnowing out process that occurs in introductory programming classes. A decrease in both interest in programming and in confidence in their aggregate ability to program had occurred, but the decrease in interest was mostly from the face-to-face class. Students were provided with a multi-part 5-point Likert scale question regarding the factors that most fostered learning in the course. In addition to the items in the first survey, a question was asked regarding the utility of the online synchronous learning sessions. Those sessions had the highest mean score (1.46/5) for importance as a contributor to learning. The factors with the next highest mean scores were linked resources and programming assignments.

While the number of survey respondents was too small for any meaningful generalization on these results, they point to some intriguing issues that warrant further study. The most interesting result was the very high rating for the online synchronous sessions as contributors to learning. Further study of this result is clearly indicated. Another important question is whether or not it is typical for introductory programming students, even ones who have had prior programming classes, to experience a decrease in interest and confidence in early classes. The opposite outcome would be desirable.

Another interesting result was that interest and confidence went hand-in-hand on both surveys. Only two of the 28 survey completers indicated a difference of more than one increment between interest and confidence. All students indicated either the same level or more interest than confidence in their abilities. A clear relationship appeared to exist between their interest and confidence measures, and final grades. It is quite likely that initial interest and confidence are strong predictors of outcome, especially in courses such as this one in which students had some prior programming experience. Additional exploration of strategies through which synchronous learning activities might bolster interest and confidence would be beneficial.

4.6. *Results of the Surveys Administered by the Academic Technology Center*

Surveys distributed to the online students midway and at the end of the course provide interesting data with regard to course design and delivery. Eleven students completed the mid-course survey and 3 students completed the end-of-course survey. The mid-term surveys consisted of demographic, background, 10 Likert-scale, and 3 open-ended questions regarding course experiences. The results showed that 90% of the online students agreed or strongly agreed that they had the technical competence necessary to succeed with online courses. Also, 90% of the students agreed or strongly agreed that (a) ongoing communication with the instructor was necessary for success in the course, (b) the level of interaction with the instructor of this course contributed to their understanding of course objectives, and (c) they were comfortable with the course management system used for the course. Additionally, 80% of the students agreed or strongly agreed that they (a) normally received responses to their emails within 24 hours, (b) normally received assignment feedback within one week of submitting assignments, and (c) had sufficient opportunity to interact with other students online. Also, 70% of the students agreed or strongly agreed that (a) the online orientation provided useful information, (b) they received constructive assignment feedback, and (c) the level of interaction between students contributed to their understanding of course objectives.

Seven students answered an open-ended question that asked if they were receiving the amount of support needed to be successful online learners. All of the responses indicated they had, although one student stated that support was more difficult for students who work “second shift”. Six students answered an open-ended question addressing what they would change about the course so far. Three students stated that they would not change anything. Two students stated that they were dissatisfied with the textbook. One student indicated that it appeared as though less teaching effort was given to the online course compared to face-to-face courses.

The survey distributed at the end of the course consisted of demographic and background questions, 46 Likert-scale questions regarding the online learning experience, course content, course structure, course appeal, course technology and support, the instructor, the learning environment, and 3 open-ended questions. Only three students submitted the end-of-course survey, however the results are quite interesting. Responses to the question that asked students if they would have preferred to take the course in a face-to-face environment, one student agreed, one was neutral, and one disagreed. Yet, all three students agreed or strongly agreed that they liked to learn online, and agreed or strongly agreed that they would take another online course. All responses regarding the online learning experience were favorable or neutral.

In responses to 13 questions regarding course content, all three students agreed or strongly agreed that (a) the course content was clear, understandable, and aligned to the course objectives, (b) the assignments were clear, relevant, and challenging, and (c) they learned a great deal in the course. With regard to course structure, all three students agreed or strongly agreed that (a) the course learning outcomes/objectives were clear, (b) the course material was well-organized, (c) the directions for the course were clear, (d) the structure of the course was easy to understand and follow, and (e) the instructional guides were helpful in focusing on the important topics.

All three students were neutral, agreed, or strongly agreed that (a) the course was accessible when needed, (b) they did not have any technology-related problems, (c) the course management system was easy to use, (d) the online course discussions, drop box, quizzes, and grade book were easy to use, and (e) the help disk assisted them in solving technical problems. All three students agreed or strongly agreed that (a) they interacted often with their instructor, (b) they received constructive assignment feedback, and (c) the instructor responded to questions.

Responses to open-ended questions included one suggestion for more video within the course content, and one suggestion for more specific assignment instructions. All three students were neutral, agreed, or strongly agreed that (a) the course was interesting and that they were satisfied with the course. All three students agreed or strongly agreed that they would recommend the course to other students. The survey results reflect favorably on the online course design and implementation, and informally validate conclusions drawn from our comparison of data between the face-to-face and online sections of the foundational programming course.

4.7. *Analysis of Student Learning Outcomes*

An analysis of student learning outcomes for the course was performed in an effort to measure relative achievement of students in the face-to-face class ($N = 19$) compared to students in the online class ($N = 16$). Descriptive measures of the two groups are summarized in Table 3.

An independent means t-test was performed on the final grades achieved by the students in the two groups. Levene’s test for equality of variance showed no difference in

Table 3

Descriptive statistics for the online and face-to-face classes

Group	***N***	**Final Mean GPA**	**Standard Deviation**	**Standard Error of the Mean**
Online	16	2.63	1.42	.355
Face-to-face	19	2.73	1.26	.289

Table 4

Results of an independent means t-test of final grades for the online and face-to-face classes

	t	**df**	**Significance (2-tailed)**	**Mean Difference**
Mean Grades	−.235	33	.816	−0.107

the dispersion of grades ($F = .492$ and $p = .488$). The results of the t-test are presented in Table 4, which indicate that no significant difference was found in the attainment of the two groups.

Although the face-to-face class achieved a slightly higher final average grade than the online class, very little difference (slightly more than 0.1%) between the means existed in this study. As previously mentioned, both classes had to complete the same programming projects and exams. All graded items were graded by the same teaching assistant (programming projects) or instructor (exams), supporting a consistent evaluation of student learning in both groups. This result is mediated by the fact that, as noted previously (El-Sheikh *et al.*, 2007), the attrition rate was higher in the online version of the course compared to the face-to-face version. Specifically, 3 students withdrew from the face-to-face course (a 14% attrition rate) compared to 13 students who withdrew from the online version (45% attrition).

4.8. *Results of the End-of-Semester Course Evaluations*

Students in both sections were asked to complete state-mandated course evaluation forms at the end of the semester. The forms asked students to answer 18 questions related to the course organization and instructor's teaching skills using a 5-point Likert scale indicating responses of *Excellent*, *Very Good*, *Good*, *Fair*, or *Poor*. Ten out of 19 students (53% response rate) completed the course evaluation forms from the face-to-face section and 6 out of 16 students (38% response rate) from the online section, which are typical response rates for such forms. A comparison between the face-to-face and online sections of the percentage of students answering *Excellent* or *Very Good* to various questions is given in Table 5.

Overall the results were encouraging, with a majority of students in both sections responding favorably (*Excellent* or *Very Good*) to all questions. Ratings provided by face-to-face students were somewhat higher than those provided by online students for the majority of the questions.

On the question regarding the facilitation of learning, 90% of face-to-face students rated the facilitation of learning as excellent or very good, with 60% of students rating it as excellent. In comparison, approximately 67% of online students indicated that the facilitation of learning was very good or excellent, with half of those students rating it as excellent. This result suggests that it is possible to engage students in an online programming course and to identify appropriate ways to facilitate learning in an online environment. In alignment with this result, approximately 67% of online students also indicated that stimulation of interest in the course, availability to assist students, and

Table 5

Percentage of students responding *Excellent* or *Very Good* to selected course evaluation questions in the face-to-face and online classes

Assessment Question	**Face-to-Face Section**	**Online Section**
Facilitation of learning	90%	66.7%
Stimulation of interest in the course	80%	66.7%
Communication of ideas and information	88%	66.7%
Availability to assist students in or out of class	80%	66.7%
Class meetings	100%	83.4%
Course requirements	100%	100%
Course assignments	89%	83.4%
Expression of performance expectation	90%	66.7%
Overall assessment of instructor	100%	66.7%
Overall course organization	100%	66.7%

communication of information was very good or excellent, compared to 80% of face-to-face students for the first two items, and 88% for the third item. This also suggests that online students felt that they were able to understand the ideas and information presented through online delivery, get assistance if they needed it in such an environment, and maintain an interest in the material.

Another significant result from analysis of the course evaluation forms was that 100% of online students rated the class meetings favorably with respect to their usefulness, with 66.7% rating them as excellent, 16.7% rating them as very good (giving a total of 83.4% shown in the table), and 16.7% rating them as good. Class meetings for the online students were the weekly synchronous sessions that were conducted using Elluminate. This result suggests that students found the synchronous sessions useful towards the facilitation of their learning. It is interesting to note that 100% of the face-to-face students also indicated that their class meetings were useful.

All students who completed the evaluation forms from both sections rated the course requirements as excellent or very good. With respect to the usefulness of the course assignments, 83.4% of online students rated the assignments as excellent or very good, which is very comparable to the 89% rating provided by face-to-face students. In addition, 66.7% of online students indicated that the expression of performance expectations for the course was very clear, compared to 90% of face-to-face students. Collectively, these results are important because they suggest that both online and face-to-face students felt that they had similar learning experiences with respect to the class meetings, course requirements, and assignments.

Although the results for the overall assessment of instructor and overall course organization appear to indicate a difference between the face-to-face and online students, a closer investigation reveals a few subtle similarities. 66.7% of online students rated the overall assessment of the instructor as excellent, compared to 60% of face-to-face students. In addition, 66.7% of online students rated the overall course organization as excellent, compared to 80% of face-to-face students. The parallel between these ratings further suggests that the online and face-to-face students had comparable learning experiences in the course.

5. Conclusions and Discussion

Adapting foundational programming courses for online delivery is a challenging task, but an essential one to solve effectively, given the increasing demand for online courses and programs. The purpose of the work reported here was to employ a variety of technologies, methods, and materials in an online version of an introductory computer programming course, and to assess the outcomes that resulted. The remainder of this section contains conclusions regarding these issues.

With regard to /research question 1, the best mix of technologies, methods, and materials, several conclusions might be drawn. A wide mix of technologies and methods were utilized in the course, and with a few notable exceptions, provided support as expected. However, a few surprises occurred. For example, discussion forums are an integral part of any online course. This version of the course included a discussion forum for "muddiest points". It was hypothesized that this particular forum would generate useful synchronous and asynchronous discussions. However, it generated very little discussion, while other discussion areas that directly related to graded course elements, such as programming projects or exams, generated a considerable amount of ongoing discussion throughout the semester. In an online course, students have a variety of competing means through which to gain clarifications on difficult-to-understand points, possibly accounting for this result. Despite this surprising outcome regarding a potentially interesting course component, discussion forums generally will remain an important part of the course materials.

The incorporation of weekly online synchronous sessions was novel to the delivery of the online programming course. These sessions provided an opportunity for students and the instructor to interact directly and synchronously using text, audio, and shared documents and applications to discuss course-related concepts and assignments. Survey results and feedback strongly suggested that the synchronous sessions were helpful for both online and face-to-face students. However, a surprisingly small number of students actually participated in the sessions. It appears likely that, based upon informal comments from students, many more benefited from the sessions indirectly by viewing the session recordings. Collection and analysis of more detailed data regarding usage patterns of the recordings, data that was not available for the current study, would provide a better picture of the broader benefits of online synchronous sessions. Future work will also include the evaluation of course resources through correlation with learning outcomes and assessment measures.

The second research question pertains to outcomes in the course. An encouraging result of this study is that very little difference was noted in attainment between the face-to-face and online sections of students completing the course. However, the online students clearly had a higher attrition rate. It should also be noted that the attrition rate in the online course was the same rate reported for the first iteration of the course. The online synchronous sessions were anticipated to aid retention in the course. Given the relatively low participation in the synchronous sessions by online students, it is likely that the full potential of the synchronous sessions is yet to be realized. The higher attrition rates evidenced in the online course compared to the face-to-face version are certainly not unique to computing disciplines (Angelino *et al.*, 2007; Carr, 2000). Angelino, Williams, and Natvig report that attrition rates for online courses are typically 10–20% higher than for face-to-face courses. Carr suggests that potential online students should be asked about their self-responsibility and computer literacy skills before the course begins. A major goal of future work must be to address the retention issue.

Comparison of results between the face-to-face and the online sections suggests that more required interactions and attendance may be necessary to improve student success in an online programming course. Online students had a clear tendency to skip events unless they were required. Decline in student participation in both face-to-face and online activities suggests the need to increase the number of required activities in both delivery formats. Institutional policy does not currently support the requirement for students to engage in virtual synchronous activities in online courses. This is due to the fact that online courses do not have a time slot associated with them. Despite efforts to identify a best time for everyone, the scheduling of synchronous events is challenging. Modification of institutional procedures and policies that better support and enforce such online synchronous learning activities would help.

The learning outcomes and attrition rates must be viewed in another light – the substantial differences in background among the participants in the course. Survey results revealed significant differences in prior programming experience between the online and face-to-face sections. No students in the online section had any prior Java programming experience. This fact certainly had an effect on successful completion rate in the online course. The survey results also pointed to an interesting relationship between interest and confidence measures, and final grades. Based on an informal analysis of the results, it appears that initial interest and confidence are strong indicators of learning outcomes in introductory programming courses. These initial results warrant further investigation. Longitudinal studies that track online and face-to-face students' interest level and confidence in programming skills as they progress through the program would provide useful data.

Although this study was designed to evaluate the effectiveness of various technologies, materials, and methods used to support online learning with the goal of informing best practices related to teaching foundational programming online, the infrastructure at our institution for online courses permits access by face-to-face students as well. Access to the online synchronous sessions was provided to the face-to-face students partway through the semester to provide additional opportunities for collaborative learning and student engagement. A small proportion of face-to-face students did choose to make use of this optional resource. Further consideration regarding the benefit of overlaying technologies originally designed for online students into face-to-face environments is needed. Course delivery strategies that integrate best practices for face-to-face and online learning clearly appear to be worthy of further study.

References

Angelino, L.M., Williams, F.K. and Natvig, D. (2007). Strategies to engage online students and reduce attrition rates. *The Journal of Educators Online*, **4**(2).

Bowers, M. (2007). Groupwork activities in synchronous online classroom spaces. In *Proceedings of SIGCSE 2007*: *The 38th ACM Technical Symposium on Computer Science Education*. Covington, Kentucky. ACM Press, New York, 91–95.

Carr, S. (2000). As distance education comes of age, the challenge is keeping the students. *The Chronicle of Higher Education*, **23**, A39.
`http://chronicle.com/free/v46/i23/23a00101.htm`

Desire2Learn Innovative Learning Technology (2008).
`http://www.desire2learn.com/`

Elluminate Live eLearning Solutions and Collaboration Solutions Software (2008).
`http://www.elluminate.com/`

El-Sheikh, E., White, L.J. and Coffey, J.W. (2007). Reflections in uncharted waters: teaching foundational programming online. In *Proceedings of EISTA 2007*: *The Fifth International Conference on Education and Information Systems, Technologies and Applications*. Orlando, FL July 12–15, 2007, 310–314.

Fisher, G. and von Gudenberg, J.W. (2006). Improving the quality of programming education by online assessment. In *Proceedings of PPPJ 2006*: *Principles and Practices of Programming in Java*, Manheim, GA. ACM Press, New York, 208–211.

Kleinman, J. and 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*, **17**(6), 206–219.

Molstad, L. (2001). Teaching computer programming using distance education technology. *Journal of Computing in Small Colleges*, **17**(1), 265–277.

Reeves, T., Baxter, P. and Jordan, C. (2002). Teaching computing courses – computer literacy, business microcomputer applications, and introduction to programming online utilizing WebCT. In *Proceedings of the 11th Annual Rocky Mountain Conference of the Consortium for Computing Sciences in Colleges*. Farmington, NM, 290–300.

Thomas, R. (2000). Experiences teaching C++ online. *Journal of Computing Sciences in Colleges*, **15**(5), 214–222.

Ury, G. (2004). A comparison of undergraduate student performance in online and traditional courses. *Journal of Computing Sciences in Colleges*, **19**(4), 99–107.

Zachery, J.L. and Jensen, P.A. (2005). Exploiting value-added content in online course: introducing programming concepts with HTML and JavaScript. In *Proceedings of SIGCSE 2005*: *The 36th ACM Technical Symposium on Computer Science Education*. Reno, NV. ACM Press, New York, 396–400.

E.M. El-Sheikh is an assistant professor in the Department of Computer Science at the University of West Florida. She currently serves as the instructional strategies and assessment fellow with UWF's Center for Teaching, Learning, and Assessment, and as an administrative fellow in the College of Arts and Sciences Dean's Office. Dr. El-Sheikh earned her PhD (2002) and MSc (1995) in computer science from Michigan State University and her BSc (1992) in computer science from the American University in Cairo. She has taught a variety of undergraduate and graduate computer science courses, including several online courses. Her research interests include the development and application of artificial intelligence-based techniques and tools for education, including the development of intelligent tutoring systems and adaptive learning tools, and research on computer science education.

J.W. Coffey holds a BS in psychology from the College of William and Mary (1971), a BS in systems science (1989), an MS in computer science/software engineering (1992), and an EdD with an emphasis in computer science (2000) from the University of West Florida (UWF). Since 1992 he has divided his time between the Department of Computer Science at UWF where he currently holds an associate professor position, and the Institute for Human and Machine Cognition (IHMC). His research interests include advanced technology for education, knowledge elicitation and representation, student modeling and web services.

L.J. White served on active duty in the U.S. Navy for 21 years. She is now an assistant professor in the Department of Computer Science at the University of West Florida. Dr. White earned a BS (1984) in computer engineering from the University of New Mexico, a MS in computer science from the Naval Postgraduate School, and a PhD (2006) in instructional design for online learning from Capella University. Her research interests primarily include software engineering, and online collaboration and learning.

Technologijų, išteklių ir metodų, reikalingų internetinio programavimo pagrindų kursui, analizė

Eman M. EL-SHEIKH, John W. COFFEY, Laura J. WHITE

Kompiuterinio programavimo kursai sudėtingi ir sunkūs dėl įvairių priežasčių. Auganti internetinio pristatymo paklausa, efektyvios technologijos, ištekliai ir metodai, geriausiai išreiškiantys interneto ryšius su mokslu, atlieka svarbų vaidmenį – padeda besimokančiajam siekti geriausių rezultatų. Šiame straipsnyje aprašomi Floridos universitete neseniai atlikti tyrimai, kuriais buvo siekiama pagerinti Java programavimo pagrindų kurso projektą ir internetinį pristatymą. Internetiniu kursu, sudarytu iš įvairių technologijų ir išteklių bei sinchroninių komponenčių sąveikos internete, norima padidinti besimokančiųjų pažangumą – tai analogiškas telekonferencijos modelis. Naudojant įvairius vertinimo metodus buvo atlikti studentų kompetencijų, perspektyvų, internetinio kurso išvadų ir etaloninio testo palyginimai. Gauti rezultatai parodė, kad internetinės sinchroninės sesijos ir kai kurie kiti kurso aspektai didina internetinio mokymosi efektyvumą. Straipsnyje taip pat aprašomi tyrimo rezultatai, pateikiamos išvados, keliamos problemos, reikalaujančios tolimesnių nagrinėjimų.