Handbook of Distance Learning for Real-Time and Asynchronous Information Technology Education

Solomon Negash
Kennesaw State University, USA

Michael E. Whitman
Kennesaw State University, USA

Amy B. Woszczynski
Kennesaw State University, USA

Ken Hoganson
Kennesaw State University, USA

Herbert Mattord
Kennesaw State University, USA
Chapter XIV

A Hybrid and Novel Approach to Teaching Computer Programming in MIS Curriculum

Albert D. Ritzhaupt
University of North Florida, USA

T. Grandon Gill
University of South Florida, USA

ABSTRACT

This chapter first discusses the opportunities and challenges of computer programming instruction for management information systems (MIS) curriculum, which includes the development of survey instruments and the meaningful integration of information and communication technology. Second, the chapter describes a unique and hybrid computer programming course for MIS curriculum that embraces an assignment-centric design, self-paced assignment delivery, low involvement multimedia tracing instructional objectives, and online synchronous and asynchronous communication. Third, the development and use of a survey is employed as a method to monitor and evaluate the course, while providing an informative discussion with descriptive statistics related to the course design and practice of computer programming instruction. Tests of significance show no differences on overall student performance or satisfaction using this instructional approach by gender, prior programming experiences, or work status. This chapter aims to provide generalizable knowledge to influence the practice in computer programming instruction in MIS curriculum.

INTRODUCTION

Graduates of management information systems (MIS) programs should possess a variety of organizational and technical skills, including a strong foundation in computer programming. While a majority (78%) of 1,250 information technology managers surveyed in a large national study suggested full-time study as the most effective way to gain the necessary skills and knowledge, only 20% of this same group reported that undergraduates were “equipped for work” (Brandon, Pruett, &
A Hybrid and Novel Approach to Teaching Computer Programming in MIS Curriculum

Wade, 2002). Research sponsored by the National Science Foundation (NSF) has reported that the U.S. has an inability to generate well-prepared new graduates in the information systems-centric disciplines (Lidtke, Stokes, Haines & Mulder, 1999). The study found that graduates lack computer programming knowledge and skills necessary to succeed in business and industry. These reports are of paramount concern to MIS educators and a signal to improve the quality of instruction in our programs.

To heighten the quality of instruction, educators place emphasis on the development and dissemination of best practices that intersect pedagogy, content domain, and information and communication technology (ICT) for instruction. Publication venues like the Journal of Information Systems Education or the Journal of Information Technology Education collect and share experiences about pedagogy in information-centric programs. These best practices are, perhaps, the necessary elements to equip our educators, and consequently, our graduates to compete in a global economy. This shared value highlights the importance of a Handbook of Distance Learning for Real-Time and Asynchronous Information Technology Education.

In the spirit of this important tradition, this chapter addresses the concerns by: 1) providing a rich description of the pedagogical context used in a novel, hybrid computer programming course in an MIS curriculum; 2) providing empirical evidence that demonstrates the instructional value of those elements found within this course; and 3) providing reliable and valid evidence of an instrument designed to monitor this course. The chapter first briefly examines the challenges and opportunities of teaching computer programming in MIS curriculum, and then examines the specific course under investigation. This chapter aims to provide generalizable knowledge to influence the practice in computer programming instruction in MIS curriculum.

The Challenges

Computer programming instruction in MIS curriculum poses many serious problems to educators, starting with the inherent difficulty of the content domain. Computer programming students have to learn to analyze problems critically, implement robust solutions in a programming language, debug code, and make enhancements to existing computer programs, and repeat this process several times in multiple programming assignments over the duration of a quarter or semester. All of this must be done while learning programming concepts, a programming language, and principles of software design. There is little surprise students are often challenged by one or more aspects of a computer programming course.

Empirical studies confirm that students struggle with computer programming. The most troubling numbers are from the introductory computer programming courses where failure and withdrawal rates exceed 50% (Woszczynski, Guthrie, & Shade, 2005). One study found that the probability of passing an introductory undergraduate programming course the first time was 40% across all majors, with an initial failure rate of 19.5% and a withdrawal rate of 40.5% (Beise, Myers, VanBrackle, & Chevli-Saroq, 2003). During a period of high enrollment growth, this may not have been such a problem. However, during a period of low enrollment, this problem can threaten the sustainability of an academic program.

Graduates of MIS programs are required to possess a strong foundation in computer programming. Yet, research suggests the degree of interest in learning computer programming is highly variable in MIS curriculum because many graduates pursue careers in the field where computer programming is not a required job activity (Gill, 2005a). In opposition to degree programs like computer science, with a more computer programming focused curriculum, MIS students may only be exposed to a single programming course in their entire program of study.
To add to the complexity of computer programming instruction, there is also increased emphasis on offering access to higher education at a distance. Taking classes at a distance poses a different set of challenges for students who are used to taking on-campus classes in terms of studying, time management, and autonomy (Moore & Thompson, 1998). More troubling, educational research on distance learning suggests that retention rates tend to be significantly lower in distance education classes (Carr, 2000; Garrison, 1987; Zajkowski, 1997).

Finally, developing MIS curriculum is an ongoing, dynamic process. Government and industry needs, along with fundamental changes in information technology are constantly driving the field of MIS to revise curricula (Al-Rawi, Lansari, & Bouslama, 2005). For instance, in 1987, Microsoft released Visual Basic 1.0 ©. This was followed by five more major releases of the Visual Basic language © (version 2.0 – 6.0). In 2001, Microsoft then released the Visual Basic .NET © language, which transformed the language into a fully object-oriented programming language. Subsequently, the company issued two more major releases of the product and associated .NET framework. Even before MIS educators are fully comfortable with one technology, they have to change it to keep up with practice.

The Opportunities

Due to the complex nature of computer programming instruction in MIS curriculum, a door of opportunity opens for educators and researchers to address the challenges and impact of both MIS educational research and practice. Two dimensions come to light in facing these challenges: 1) the integration of ICT to heighten the quality of instruction in face-to-face, distance and hybrid modality, and 2) the development, validation, and use of instruments designed to measure a number of factors influencing a student’s perspective of a course. Both of these opportunities intersect in the goal of improving instruction.

The integration of ICT into MIS curriculum for instructional purposes is a clear fit for the discipline, since MIS faculty tend to be knowledgeable both about how to use ICT and the process surrounding its integration. ICT speaks of the infusion of tools to store, retrieve, and manipulate information with tools for communication. Integration initiatives range from using multimedia CD-ROM course delivery (Doube, 1998) to fully online instruction using specially developed course management systems (Molstad, 2001) to hybrid courses combining both face-to-face and distance technologies (Gill, 2006). The initiatives also integrate a number of different instructional strategies, like language independent approaches or cooperative learning (Lehman & Naumann, 1986; Nosek, 1998; Williams & Kessler, 2001). The opportunities are endless, yet the empirical characterization of these initiatives should demonstrate student achievement and satisfaction equal to or above traditional methods (Stansfield, McLellan, & Connolly, 2004).

In speaking of student satisfaction, the primary means that most instructors have for assessing their courses are word-of-mouth and university-wide course evaluations. While these two forms of feedback are helpful, they tend to be either anecdotal or very general in nature. To gain the sufficiently detailed information necessary to evaluate individual course elements, MIS educators may need to develop and tailor instruments to capture information that directly pertains to their courses.

Of course, the development of instruments to assess course design and student perspectives immediately raises concerns about reliability and validity. The development, validation, and use of instruments to assess these areas therefore present a fruitful research opportunity. As pointed out in the literature (Straub, 1989), confirmatory empirical findings will be strengthened in information systems research when “instrument validation precedes both internal and statistical conclusion validity” (Straub, 1989, p. 147).
MIS PROGRAMMING COURSE

The programming course for which this research is based is offered at a large university ("very high research activity") in a metropolitan city in the southeast United States (Carnegie Classification, 2007). The course is a required introductory programming course for undergraduate MIS majors, taught using the C++ programming language. The course is generally taken during a student’s junior year, and is one of the first courses taken in the MIS major. The course historically enrolls anywhere from 80 to 100 students in the spring and fall semesters. The purpose of the course is to teach all students the basics of procedural and object-oriented thinking so students can pursue any career in MIS. The course used to be delivered in a face-to-face format, but later was transformed into a hybrid format, incorporating many different technologies and pedagogical strategies. Lectures were entirely removed from the course.

The course curriculum includes a variety of conceptual, algorithmic, and practical elements, such as data representation, flowcharting, functions, elementary algorithms, debugging techniques, memory organization, and input/output streams. Students are also introduced to classes, polymorphism, encapsulation, and inheritance. The course was supported by one instructor and up to five teaching assistants in any given semester. Table 1 shows the seven programming assignments students should complete in this course. The course description is:

**Business Application Development**—**Presentation of business application development using a modern programming language.** Topics include data structures, indexing, file processing, and user interfaces. Good program design techniques are emphasized. Business applications are developed.

The course diverges from traditional programming courses in that it is essentially self-paced, heavily supplemented with multimedia materials, assignment-centric, and makes extensive use of both synchronous and asynchronous communication technology to support the learning process. These characteristics should not be viewed as independent. Rather, they are part of an integrated approach with the distinct elements intersecting at many different levels. The follow-

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment 1: Compiler Exercises</td>
<td>Compiler installation and simple compiles (Hello, World! and simple multfile project).</td>
</tr>
<tr>
<td>Assignment 2: Numbering Systems</td>
<td>Conversions between decimal, hex, and binary. Twos complement representation. Simple bitwise logical operations. Credit for assignment will be dependent on the results of an online exam conducted in the lab.</td>
</tr>
<tr>
<td>Assignment 3: Logic and Flowcharting</td>
<td>Creating flow charts for simple processes. Converting code to flow charts. Converting flow charts to code. Credit for assignment will be dependent on the results of an oral exam.</td>
</tr>
<tr>
<td>Assignment 4: Debugging and Pointer Arithmetic</td>
<td>Taking a program with a variety of compiler, linker and runtime errors and finding/removing the bugs. Using a memory grid to locate items in memory. Credit for assignment will be dependent on the results of an online exam conducted in the lab.</td>
</tr>
<tr>
<td>Assignment 5: Function Exercises</td>
<td>Creating a series of functions that perform simple string tasks. Credit for assignment will be dependent on the results of an oral exam.</td>
</tr>
<tr>
<td>Assignment 6: Structured CGI Application</td>
<td>Creating Web-based application that takes input from a Web form and returns it to a browser. Credit for assignment will be dependent on the results of an oral exam.</td>
</tr>
<tr>
<td>Assignment 7: OOP CGI Application</td>
<td>Rewriting Web-based CGI application using C++ classes. Credit for assignment will be dependent on the results of an oral exam.</td>
</tr>
</tbody>
</table>
Assignment-Centric Design

Assignment-centric design can be characterized by the following principles: completion of the course assignments meets the course requirements; all assessment is directed toward validating students properly to complete the assignments; and the primary role of instruction is helping students complete the assignments (Gill, 2005a). This approach is similar to a project-driven approach, in which the programming assignments are designed to trace all learning objectives in the curriculum and drive the instruction in the course (Ritzhaupt & Zucker, 2006). However, traditional quizzes and examinations are removed from the curriculum, leaving only programming assignments in place with assessment directed at validating student completion of the assignments.

The integration of an assignment-centric design requires substantial modifications to a traditional course. These changes include providing access to learning materials in a flexible format, directing lab sessions towards assignment completion, and providing technical support 7-days a week to accommodate diverse schedules (Gill, 2005a). A Web-based content delivery system was developed combining both a course management system (Blackboard©) for asynchronous communication and a Web site with all the instructional materials organized in assignment modules.

Validation of the exams involves two methods: 1) proctored online exams supervised by teaching assistants; and 2) oral exams with either a teaching assistant or instructor. The oral exams include specific questions to probe whether students completed and understood the material they submitted. Questions are typically open-ended, such as “What does this line of code do?” or “What would happen if this line of code were removed?” Any code included in an assignment submission is open to inclusion in the oral exams (Gill, 2005a).

Two other important characteristics surrounding the validation of an assignment include: 1) having no specific limit on the number of attempts a student can make on validating the assignment; and 2) the degree of the validation being directly proportional to the performance on the assignment. “Thus, a 95% score on an assignment meant a far tougher validation exam than a 60%” (Gill, 2005a, p. 342).

Self-Paced Delivery

This course embraces a self-paced approach. The development of a self-paced approach included the integration of three different, interdependent systems: a content delivery system, peer support system, and progress monitoring system. The content delivery system, as previously noted,
A Hybrid and Novel Approach to Teaching Computer Programming in MIS Curriculum

included a Web site with all instructional materials, a course management system for group discussions, and some validation assessments, and later included an Elluminate Live!© component for conducting online lab sessions and a virtual office (Gill & Holton, 2006).

The peer support system was implemented by encouraging students to cooperate with each other, and teaching assistants (all of whom had previously completed the course as students) holding office hours and answering questions posted to the discussion board or via e-mail. Though traditional programming classes do not allow students to cooperate on their assignments, in the real-world, software development is rarely a solitary activity. Peopleware (DeMarco & Lister, 1999) reports that software developers generally spend 30% of their time working alone, 50% of their time working with one other person, and 20% of their time working with two or more people.

Cooperative strategies like pair-programming as an instructional method have seen tremendous growth in computer programming courses (Nosek, 1998; Williams, & Kessler, 2001). At the same time, the obvious drawback of cooperation is the potential loss of academic rigor that can result from the “free rider” effect (Gill & Holton, 2006). In practice, the assignment validations address this problem because students completing an assignment receive no credit for doing so until the assignment is validated.

The progress monitoring system was modeled after nuclear submarine training (Gill, 2005b). Each student was provided a validation card for progress monitoring (validation assessments and programming assignments) and was paired with one teaching assistant. Further, a system of credit for each assignment was instituted in which students had to update their assigned teaching assistant on their progress and post to an asynchronous, online progress report form. These various elements worked in concert to encourage students to complete the activities at an even pace (Gill & Holton, 2006). While the self-paced approach allowed more flexibility for students, the instructor did have suggested turn-in dates for each assignment and hard deadlines specifying when all assignments would be due, typically right before the exam period at the institution.

Low Involvement Multimedia

Developing high quality multimedia resources for distance learning, face-to-face, or hybrid courses is a resource intensive process, drawing from faculty time and institutional monies. This high-involvement process can quickly serve as a deterrent to faculty members with split research, teaching, and service responsibility. Consequently, there is a need for user-friendly, low budget multimedia authoring tools. This has led to the development of authoring tools like Camtasia© (2007) and Articulate© (2007), and explains their explosive growth in the higher education market.

For this course, multimedia tutorials have been developed and mapped to the instructional objectives of each assignment by the instructor. The multimedia materials included animated screen captures and presentations with the instructor’s voice as guided narration (Gill, 2007). The primary advantage of these multimedia resources to both students and instructors is that a tremendous amount of information can be communicated in a relatively short amount of time. Resources can be developed as planned lectures or on-the-fly productions to answer student questions (Gill, 2007).

The multimedia resources developed using these authoring tools can be seamlessly integrated into a course management system or simply uploaded to Web space for efficient delivery. The resources could be burned to CDs and distributed early in the semester (an approach previously used in this course). More recently, the pervasiveness of broadband Internet connections has made
A Hybrid and Novel Approach to Teaching Computer Programming in MIS Curriculum

Web-based delivery the near-universal choice of students.

A critical lesson from the experiences of this course is that faculty members do not have to be multimedia development specialists to wield this powerful technology for their own activities. Given minimal support and tools, faculty members can generate materials for entire courses with a few months of preparation and implementation. Often, academic institutions will also provide support personnel to aid faculty in this development process.

Synchronous Online Interaction

One of the unfortunate drawbacks to most forms of multimedia is that students cannot ask questions and receive immediate feedback. Until recently, higher education did not have access to robust and scalable tools for synchronous, online communication aside from basic chat rooms and whiteboards without audio capability. Over the past 5 years, we have seen tremendous market growth in tools like Adobe Breeze© or Elluminate Live! to support online, synchronous interaction. However, synchronous communication tools like Elluminate Live! have not been thoroughly investigated in research literature (Johnson, 2006) and are often cost-prohibitive, which complicates the integration process.

Elluminate Live! is a voice-over Internet protocol (VoIP) package that has been particularly successful in penetrating the higher education market. The software package has many features making it well-suited for computer programming instruction. Of particular note in this regard is application sharing. This feature allows the instructor to give live demonstrations of writing source code, compiling, debugging, and even running programs. The students see exactly what the instructor sees and hear exactly what the instructor says. Students can ask the instructor questions during live demonstrations and receive immediate feedback as the tool is built to emulate a virtual classroom environment.

Elluminate Live! was not integrated into the course until the spring of the 2004-05 school year because the software was previously unavailable to the institution. The software package was used by the instructor and teaching assistants to provide virtual office hours and to hold online lab sessions to demonstrate skills and knowledge specific to assignments, such as writing functions. Students were not required to join the sessions, but rather the sessions served as supplementary instructional support. Additionally, those students that could not attend an online session could view the recorded session later in an asynchronous modality.

Course Grading Scale

The grading scale, presented in Table 2, was fixed throughout the study, although the weight given to individual assignments and participation did change over time. Because of the assignment-centric approach, the instructor instituted a system in which only the first four assignments had to be satisfactorily completed in order to receive a C grade, the first five assignments for a grade of a B, and six of the seven assignments to receive an A grade. Completing the requirements of an A grade resulted in depth of coverage far beyond what would normally be expected in an introductory computer programming course.

Table 2. Grading scale in summer of 2004

<table>
<thead>
<tr>
<th>Numeric grade range</th>
<th>Letter grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>80-100</td>
<td>A</td>
</tr>
<tr>
<td>60-79</td>
<td>B</td>
</tr>
<tr>
<td>40-59</td>
<td>C</td>
</tr>
<tr>
<td>20-39</td>
<td>D</td>
</tr>
<tr>
<td>&lt;20 F</td>
<td>F</td>
</tr>
</tbody>
</table>
METHOD

Instrumentation

The primary instrument used in this analysis was developed to monitor this complex programming course, and to collect useful information about a student’s previous background and career expectations. The instrument was developed by the instructor of the course and contains over 300 items organized into five separate sections: student background information, assignments, opinions, student assessment of learning gains, and (added in spring 2005-06) Elluminate usage. The satisfaction and assessment of learning gains sections were, in turn, derived from three NSF-sponsored instruments, the “Student Opinion Survey,” the “Computer Programming Survey,” and the “Student Assessment of Learning Gains (SALG).”

The university-wide course evaluation instrument has eight items and uses a modified Likert scale in the range of 1 to 5 (1=poor, 2=fair, 3=good, 4=very good, 5=excellent), with an additional area for free-form comments. This instrument served as a secondary instrument in the analysis. The instrument was administered concurrently with the instructor-developed instrument, and is designed to measure a student’s overall level of satisfaction with various areas of the course.

Procedure

The instrument was electronically released to students at the end of the 15th week of a 16-week semester, along with the university-wide course evaluations. Students were instructed to complete the survey in an electronic format (spreadsheet) and to e-mail the results to an administrative e-mail address that the course instructor could not access. To encourage students to complete the survey, they were provided extra credit points toward their final grades, which led to typical response rates of approximately 70%.

The instrument was not anonymous for those students desiring extra credit, which allowed for the responses to be linked to responses on the university-wide course evaluation instrument, and student grades for the course. Students are instructed that they can request a small project as an alternative to this extra credit opportunity. However, this request has not been made since the inception of this procedure.

Prior to submission of course grades, a designated departmental administrative assistant provided the course instructor with a list of students who had responded, so each student’s grade could be adjusted accordingly. Copies of the completed surveys, on a CD, were then made available to the instructor only after all course grades had been officially submitted to the registrar. This procedure was designed to prevent student concerns that their responses might be considered in assigning their final course grade. During 4 years of collecting data in this fashion, not a single complaint was raised—either to the course instructor or to the department chair—regarding the mechanics of the process.

Participants

This chapter reports on students (N=254) enrolled in the aforementioned course from the spring of the 2002-03 school year to the spring of 2004-05 school year on a semester system. The instrument’s first section, labeled student background information, included a variety of background information about the student population. Approximately 69% of the participant students were male. Ages varied with a mean of 26.16 (SD=7.48) years old. Approximately 13% of the students were non-US citizens. Forty-seven percent of the students had never taken a programming course and 59% indicated this as their first C/C++ programming experience.

Nearly 40% of the students were working part-time and 35% full-time, while enrolled in the course. Interestingly, while the majority of
the students indicated working either full- or part-time, nearly 78% also indicated full-time status as a student. Eighty-two percent of the students in the sample were MIS majors with the remaining being either other business majors or nondegree seeking students.

Of particular interest was the attractiveness of MIS-related and non-MIS-related careers to students. One item in the background section of the instrument asked how attractive a particular career was using a 5-point scale. Internal consistency for the scale is less than desirable with \( \alpha = 0.49 \). The response frequencies for these items are shown in Table 3. As can be gleaned, 50% of the students indicate computer programming as an unattractive career and 17% had a neutral response. Seventy-one percent of the students are attracted to MIS project management, shortly followed by 65% attracted to both general management and network management as a career goal.

**Data Analysis**

Recognition of the need to provide valid and reliable measures served as a key component of the analysis. Quantitative analyses of the data included descriptive analysis of response frequencies and measures of variation and central tendency, internal consistency reliability analysis (Cronbach’s alpha), an exploratory factor analyses (EFA) for the student level of satisfaction subscale, and statistical inferences using analysis of variance (ANOVA) and Pearson correlations for the data that are continuous. EFA and reliability analysis were conducted to explore the underlying structure of these data (satisfaction subscale only) and to demonstrate the reliability of the measures prior to statistical inference.

**RESULTS**

In this chapter, internal consistency reliability equal to or above a 0.7 threshold is assumed to be an acceptable measure (Nunnaly, 1978). Statistical significance is set at a 0.05 level for all statistical tests. When using 5-point Likert scales, responses and central tendency measures greater than 3 (central points) are considered favorable. Results including performance measures (assignment performance) include only those students that responded to the survey.

**Table 3. Student career expectations**

<table>
<thead>
<tr>
<th>Career area</th>
<th>M</th>
<th>SD</th>
<th>VU</th>
<th>U</th>
<th>N</th>
<th>SA</th>
<th>VA</th>
</tr>
</thead>
<tbody>
<tr>
<td>General mgmt.</td>
<td>3.65</td>
<td>1.10</td>
<td>6%</td>
<td>9%</td>
<td>20%</td>
<td>43%</td>
<td>22%</td>
</tr>
<tr>
<td>Programming</td>
<td>2.68</td>
<td>1.33</td>
<td>25%</td>
<td>25%</td>
<td>17%</td>
<td>24%</td>
<td>9%</td>
</tr>
<tr>
<td>Database mgmt.</td>
<td>3.48</td>
<td>1.14</td>
<td>6%</td>
<td>15%</td>
<td>22%</td>
<td>39%</td>
<td>19%</td>
</tr>
<tr>
<td>Network mgmt.</td>
<td>3.64</td>
<td>1.11</td>
<td>6%</td>
<td>10%</td>
<td>19%</td>
<td>44%</td>
<td>21%</td>
</tr>
<tr>
<td>Sales</td>
<td>2.44</td>
<td>1.32</td>
<td>33%</td>
<td>24%</td>
<td>18%</td>
<td>17%</td>
<td>8%</td>
</tr>
<tr>
<td>MIS project mgmt.</td>
<td>3.83</td>
<td>1.10</td>
<td>4%</td>
<td>10%</td>
<td>15%</td>
<td>41%</td>
<td>30%</td>
</tr>
<tr>
<td>CPA</td>
<td>2.09</td>
<td>1.24</td>
<td>47%</td>
<td>17%</td>
<td>21%</td>
<td>10%</td>
<td>5%</td>
</tr>
<tr>
<td>Lawyer</td>
<td>2.26</td>
<td>1.43</td>
<td>48%</td>
<td>12%</td>
<td>17%</td>
<td>13%</td>
<td>10%</td>
</tr>
<tr>
<td>Medical Doctor</td>
<td>2.00</td>
<td>1.27</td>
<td>53%</td>
<td>15%</td>
<td>15%</td>
<td>12%</td>
<td>5%</td>
</tr>
<tr>
<td>Game developer</td>
<td>3.23</td>
<td>1.40</td>
<td>19%</td>
<td>10%</td>
<td>21%</td>
<td>29%</td>
<td>21%</td>
</tr>
</tbody>
</table>

*Note. M = mean response, SD = standard deviation, VU = very unattractive, U = unattractive, N = neutral/have no opinion, SA = somewhat attractive, VA = very attractive.*

267
A Hybrid and Novel Approach to Teaching Computer Programming in MIS Curriculum

The instrument did not force students to respond to any items on the instrument, as doing so is considered poor instrument design. Because the number of surveys missing large numbers of responses was minimal (less than 10 responses to any survey item), it was decided that if a student did not respond to some items and responded to others, the data would be included in the analysis for those items where responses were provided.

Instruction Quality and Involvement

The second part of the instrument, labeled assignments, included a number of items examining the student’s perspective of the assignments, the instruction supporting the assignments, and the validation assessments of the assignments. The area also attempts to measure the involvement in each of the assignments and supporting areas. Because the course uses an assignment-centric design and greatly relies on the use of multimedia resources, this type of information is especially important.

The items response scale ranged from 1 to 5 (1=strongly disagree, 2=mildly disagree, 3=neutral, 4=mildly agree, 5=strongly agree) with a not applicable option for each assignment. The internal consistency reliability of the categorical items demonstrates high internal consistency reliability at α=.93. Table 4 illustrates the response frequencies, mean, and standard deviations by assignment for these items. Assignment 7 was not included in the analysis because only a small fraction of the students completed the assignment each semester.

When examining whether students found each of the assignments to be “a helpful learning activity,” whether “the multimedia resources related to the assignment were very helpful,” and whether students could complete the assignments “with little or no difficulty,” all the mean responses are above the central point for all six assignments. In terms of the programming assignment validations, students had favorable responses to all validation exams, both online and oral.

Understanding the amount of time a student invests into the course is also a pertinent measure. If the time a student invests in a particular assignment is too great, the instructor should modify the assignment or the supporting instruction to reduce the workload on a student. Table 5 provides student responses to the number of hours invested in the completion of each assignment. The results are clear in indicating the logic and flowcharting (Assignment 3, 18.76, \(SD=16.82\)) and structured common-gateway interface applications (Assignment 6, 17.3, \(SD=17.9\)) occupied the most amount of time in the course.

Another way to view the amount of time a student invests in the course is by activity or resource area. Students were asked to indicate the amount of time spent participating in a number of areas per week. The results, shown in Table 6, demonstrate the majority of the time students are completing assignments at approximately 11.16 (\(SD=10.1\)) hours per week, which is expected since the course embraces an assignment-centric design. The amount of time devoted to the other areas varied, but all appear to be, on average, below 4 hours per week.

One concern is that students are investing too much time into a specific area. One item stated, “The assignments required too much time,” and used a 5-point agreement scale. Students indicated a mean of 3.93 (\(SD=1.15\)), suggesting that, on average, more students agreed with this statement than disagreed. This section of the instrument also provided quality measures associated with the teaching assistants supporting the course. While these measures are vital to the evaluation of the course (and instructional support staff), they were purposefully excluded from the chapter.

Student Level of Satisfaction

The level of satisfaction subscale included 16 items under the third section of the instrument labeled opinions. The response scale for these items uses a modified Likert scale in the range
## Table 4. Assignment-centric measures

<table>
<thead>
<tr>
<th>Assignment</th>
<th>M</th>
<th>S.D.</th>
<th>SD</th>
<th>MS</th>
<th>N</th>
<th>MA</th>
<th>SA</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Assignment 1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Helpful</td>
<td>4.02</td>
<td>1.23</td>
<td>4%</td>
<td>5%</td>
<td>12%</td>
<td>36%</td>
<td>43%</td>
<td>0%</td>
</tr>
<tr>
<td>Multimedia</td>
<td>4.13</td>
<td>1.26</td>
<td>3%</td>
<td>6%</td>
<td>12%</td>
<td>24%</td>
<td>52%</td>
<td>3%</td>
</tr>
<tr>
<td>Independence</td>
<td>4.22</td>
<td>1.34</td>
<td>5%</td>
<td>6%</td>
<td>8%</td>
<td>18%</td>
<td>62%</td>
<td>1%</td>
</tr>
<tr>
<td><strong>Assignment 2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Helpful</td>
<td>3.68</td>
<td>1.40</td>
<td>5%</td>
<td>12%</td>
<td>16%</td>
<td>32%</td>
<td>34%</td>
<td>1%</td>
</tr>
<tr>
<td>Multimedia</td>
<td>3.64</td>
<td>1.40</td>
<td>5%</td>
<td>10%</td>
<td>23%</td>
<td>23%</td>
<td>32%</td>
<td>7%</td>
</tr>
<tr>
<td>Independence</td>
<td>4.02</td>
<td>1.33</td>
<td>3%</td>
<td>10%</td>
<td>10%</td>
<td>30%</td>
<td>47%</td>
<td>1%</td>
</tr>
<tr>
<td>Fairness</td>
<td>4.16</td>
<td>1.05</td>
<td>3%</td>
<td>7%</td>
<td>10%</td>
<td>31%</td>
<td>47%</td>
<td>2%</td>
</tr>
<tr>
<td><strong>Assignment 3</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Helpful</td>
<td>4.04</td>
<td>1.22</td>
<td>4%</td>
<td>6%</td>
<td>12%</td>
<td>31%</td>
<td>47%</td>
<td>0%</td>
</tr>
<tr>
<td>Multimedia</td>
<td>3.69</td>
<td>1.35</td>
<td>7%</td>
<td>10%</td>
<td>18%</td>
<td>27%</td>
<td>35%</td>
<td>3%</td>
</tr>
<tr>
<td>Independence</td>
<td>3.46</td>
<td>1.38</td>
<td>10%</td>
<td>14%</td>
<td>16%</td>
<td>32%</td>
<td>27%</td>
<td>1%</td>
</tr>
<tr>
<td>Fairness</td>
<td>4.28</td>
<td>0.95</td>
<td>2%</td>
<td>3%</td>
<td>11%</td>
<td>29%</td>
<td>50%</td>
<td>4%</td>
</tr>
<tr>
<td><strong>Assignment 4</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Helpful</td>
<td>4.09</td>
<td>1.14</td>
<td>3%</td>
<td>3%</td>
<td>14%</td>
<td>35%</td>
<td>42%</td>
<td>3%</td>
</tr>
<tr>
<td>Multimedia</td>
<td>3.78</td>
<td>1.29</td>
<td>6%</td>
<td>7%</td>
<td>19%</td>
<td>27%</td>
<td>35%</td>
<td>7%</td>
</tr>
<tr>
<td>Independence</td>
<td>4.00</td>
<td>1.21</td>
<td>3%</td>
<td>8%</td>
<td>14%</td>
<td>30%</td>
<td>43%</td>
<td>2%</td>
</tr>
<tr>
<td>Fairness</td>
<td>4.09</td>
<td>1.09</td>
<td>3%</td>
<td>5%</td>
<td>17%</td>
<td>22%</td>
<td>42%</td>
<td>12%</td>
</tr>
<tr>
<td><strong>Assignment 5</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Helpful</td>
<td>3.86</td>
<td>1.42</td>
<td>5%</td>
<td>3%</td>
<td>15%</td>
<td>26%</td>
<td>34%</td>
<td>18%</td>
</tr>
<tr>
<td>Multimedia</td>
<td>3.43</td>
<td>1.47</td>
<td>7%</td>
<td>10%</td>
<td>19%</td>
<td>22%</td>
<td>22%</td>
<td>21%</td>
</tr>
<tr>
<td>Independence</td>
<td>3.01</td>
<td>1.45</td>
<td>11%</td>
<td>17%</td>
<td>19%</td>
<td>23%</td>
<td>13%</td>
<td>17%</td>
</tr>
<tr>
<td>Fairness</td>
<td>4.05</td>
<td>1.12</td>
<td>4%</td>
<td>2%</td>
<td>16%</td>
<td>17%</td>
<td>33%</td>
<td>29%</td>
</tr>
<tr>
<td><strong>Assignment 6</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Helpful</td>
<td>3.58</td>
<td>1.71</td>
<td>4%</td>
<td>3%</td>
<td>15%</td>
<td>14%</td>
<td>24%</td>
<td>41%</td>
</tr>
<tr>
<td>Multimedia</td>
<td>3.21</td>
<td>1.71</td>
<td>5%</td>
<td>8%</td>
<td>16%</td>
<td>12%</td>
<td>16%</td>
<td>43%</td>
</tr>
<tr>
<td>Independence</td>
<td>3.01</td>
<td>1.73</td>
<td>8%</td>
<td>9%</td>
<td>15%</td>
<td>14%</td>
<td>13%</td>
<td>41%</td>
</tr>
<tr>
<td>Fairness</td>
<td>3.93</td>
<td>0.99</td>
<td>1%</td>
<td>1%</td>
<td>15%</td>
<td>12%</td>
<td>16%</td>
<td>55%</td>
</tr>
</tbody>
</table>

Note. M=Mean, S.D.=Standard Deviation, SD=strongly disagree, MD=mildly disagree, N=neutral, MA=mildly agree, SA=strongly agree, Helpful=The assignment was a helpful learning activity, Multimedia=The multimedia resources related to the assignment were very helpful, Independence=I could now complete the assignment by myself with little or no difficulty, Fairness=The oral or online validation exam I took for the assignment provided a fair assessment of my knowledge at the time.
of 1 to 5 (1=not at all satisfied, 2=somewhat dis-
satisfied, 3=neutral/don’t know, 4=somewhat
satisfied, 5=very satisfied), and also included a
not applicable option. Students were instructed,
“Indicate your level of satisfaction with the vari-
ous elements/aspects of the course that are listed”
(see Table 8).

The Cronbach Alpha for the 16-item instrument
was $\alpha=0.89$, a sign of high internal consistency
for these items and data. This particular subscale
lends itself to further investigation for these data
because of its generic nature and consistent re-
sponse scale. Consequently, this subscale is used
to demonstrate the initial process of validating a
portion of the instrument. As pointed out by Straub
(1989), instrument validation should be done prior
to drawing upon any statistical inferences.

Bartlett’s test of sphericity for these data has
a chi-square of 1225.99 (p<.001), which suggests
the intercorrelation matrix contains adequate com-
mon variance. The Kaiser-Meyer-Olkin measure
of sampling adequacy is 0.85, far above the 0.5
recommended threshold (Kaiser, 1974). Finally,
the participants to items ratio is approximately
15:1, well above the 10:1 ratio for factor analysis
suggested (Kerlinger, 1974). Thus, these data
appeared to be well-suited for EFA.

An EFA was executed using principal axis
factoring and an oblique (promax) rotation since
the factors were anticipated to be correlated. The
resulting correlation matrix showed that all items
correlated in the range of $r=.03$ to $r=.70$, as they
should since the items are all positively stated.
An investigation of the item-to-total correlations
showed that reliability would not be advanta-
geously increased with the removal of an item.

Table 7 shows the Eigenvalues and variance
explained by factor. A review of the Scree plot
showed that little variability was added after the
inclusion of the fourth factor, which explained ap-
proximately 63% of the cumulative variation. For
subsequent factors, the Eigenvalues were below
Kaiser’s criterion at 1.0. Consequently, we chose
parsimony over complexity to focus the results
on a four factor model to explain a student’s level
of satisfaction.

To uncover the underlying factors of the in-
strument, the maximum factor loadings for each
item were used to assign each item to a factor.
The factor and item statistics are shown in Table
8. The results did not exhibit a truly simple
structure, but the authors felt the loadings are
reasonable for this type of instrument. The factors
correlated in the range of $r=.39$ to $r=.60$, suggest-
ing moderately strong relationships between the
extracted factors.

The extracted factors all demonstrate ac-
ceptable internal consistency reliability with the
exception of the fourth at $\alpha = .47$. Since the items
measure elements important to the particular
course, no items were removed. The other factors’
internal consistency reliability measures were .7,
.85, and .82 for factors one to three, respectively.
Future adjustment to the scale may be necessary
to increase reliability.
A Hybrid and Novel Approach to Teaching Computer Programming in MIS Curriculum

Content Validity

To demonstrate content validity of the constructs, three MIS faculty members not involved in the course or the instrument’s design reviewed the items in each factor and were instructed to determine the underlying factor by providing a title, description, and justification for each factor and its items. These responses were analyzed for similarity in semantics and language. The analysis resulted in four meaningful dimensions for the instrument: course structure and composition, instructional materials, instructor and student interaction, and ancillary support. The items and the titled factors are also shown in Table 8.

Construct Validity

Construct validity is provided by correlating the subscale with the university-wide course evalu-
A Hybrid and Novel Approach to Teaching Computer Programming in MIS Curriculum

The instrument positively correlated $r=.70$ ($n=214, p < .001$) with the university-wide course evaluation. The correlation between the level of satisfaction and the university-wide course evaluation was anticipated, since logically, a course evaluation is a theoretical measure of student satisfaction with a course.

Relationships Among Factors

A strong relationship is present between course structure and composition, and the instructional materials ($r=.59$) provided in the course. This may indicate that the instructional materials and their alignment with the number and type of assignments are critical to a student’s level of satisfaction with the course. Further, the instructional materials, and instructor and student interaction also exhibit a strong relationships ($r=.60$), which may speak to a teacher effectively using the instructional material while interacting with students. Finally, a relatively strong relationship between instructor and student interaction, and ancillary support ($r=.43$) is present. This relationship may indicate the need for student and instructor interaction in relation to the additional support, such as tutors.

Student Assessment of Learning Gains

The fourth section of the survey is labeled student assessment of learning gains, and includes a number of items pertaining to a student’s self-assessment of their learning. Particularly, the section attempts to capture a student’s reflective assessment of particular skills and knowledge that are developed as a result of completing the course. The items in this section use two 5-point scales (1=No help, 2=little help, 3=moderate help, 4=much help, 5=very much help; and 1=Not all, 2=A little, 3=Somewhat, 4=A lot, 5=A great deal) with not applicable options. The internal consistency reliability for the items in this section is more than acceptable at $\alpha=.95$.

The course involves students in many activities. Some of these activities are shown in Table 9, along with the student responses. This item asked students “How did each of the following class activities help your learning?” The highest mean responses are teamwork in labs at 4.24 ($SD=1.36$) and hands-on lab activities at 3.95 ($SD=1.3$). These results indicate that students feel these areas are most helpful in learning the content. The discussions in section and in Blackboard appear to have the lowest positive responses at 29% and 33%, respectively. This indicates students may have not perceived asynchronous discussions as a helpful learning tool in this course, perhaps because of the lack of immediacy of the medium.

Another area of interest is a student’s perception of the influence of course structure, such as the type of assignments, on their learning. The response statistics are shown in Table 10. The mean response for the self-paced approach is 2.90 ($SD=1.37$), suggesting students may not find a self-paced approach that helpful as an in-

<table>
<thead>
<tr>
<th>Activity</th>
<th>M</th>
<th>S.D.</th>
<th>NH</th>
<th>LH</th>
<th>MH</th>
<th>H</th>
<th>VH</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discussions in sections</td>
<td>3.21</td>
<td>1.51</td>
<td>15%</td>
<td>15%</td>
<td>22%</td>
<td>18%</td>
<td>11%</td>
<td>8%</td>
</tr>
<tr>
<td>Hands-on lab activities</td>
<td>3.95</td>
<td>1.30</td>
<td>4%</td>
<td>9%</td>
<td>17%</td>
<td>21%</td>
<td>29%</td>
<td>7%</td>
</tr>
<tr>
<td>Teamwork in labs</td>
<td>4.24</td>
<td>1.36</td>
<td>4%</td>
<td>7%</td>
<td>13%</td>
<td>17%</td>
<td>33%</td>
<td>14%</td>
</tr>
<tr>
<td>Blackboard discussions</td>
<td>3.23</td>
<td>1.49</td>
<td>13%</td>
<td>18%</td>
<td>19%</td>
<td>13%</td>
<td>20%</td>
<td>4%</td>
</tr>
</tbody>
</table>

Table 9. Student assessment of activity area

Note. M=mean, S.D.= standard deviation, NH=No help, LH=little help, MH=moderate help, H=much help, VH=very much help, NA=not applicable.
structional method for this type of course. Some students find themselves procrastinating when using a self-paced approach, which attests to the importance of the instructor frequently monitoring students to encourage timely completion of the work. Sixty-nine percent of the students reported the grading scale as helpful to their learning, which suggests students have a positive response to having control, based on how much effort they are willing to expend, over their final grade.

Being a computer programming course, it is important for students to be able to assess whether they have acquired the desired computer programming skills from the course. Specifically, this item asks students, “How much has this class added to your skills in each of the following areas?” The responses in Table 11 indicate a high degree of perceived skill development with all mean responses above the central point. More than 50% of the respondents indicated the course added a lot to their ability to create flowcharts (54%), reading and understanding the C++ programming language (53%), and the ability to work effectively with others (51%). Interestingly, only 39% positively indicated the course added to their ability write C++ computer programs. This finding may be attributable to the vast array of computer programming concepts, such as flowcharts and structured problem-solving skills, covered in this course, limiting students’ syntax skills. Further, some aspects of the programming assignments involved the deciphering and modification of existing computer programs as oppose to students starting from scratch.

Initial learning in the context of a programming course is important; however, educators are more concerned with the transfer of learning, in which the skills and knowledge developed during a course can be applied in a different setting by a student (Brandon et al., 2002). Students were asked, “How much of the following do you think you will remember and carry with you into your other classes or aspects of your life?” The responses are shown in Table 12. The mean student responses to both “the core programming concepts” and “how to solve problems” are above the central point, suggesting students believe they have gained skills from the course that can be used in their future academic and professional careers. This is especially important since the goal of the course is to provide students problem-solving skills and the ability to think programmatically.

**Synchronous Software Usage**

During the semester Elluminate Live! was integrated, the software had been quickly accepted by students as it was offered as a supplementary resource for virtual office hours and online lab sessions to support students in completing their programming assignments. Though the use of the software was not a requirement in this semester, 72% of the students in semester attended at least one synchronous session.

<table>
<thead>
<tr>
<th>Table 10. Student assessment of structure composition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Organization</strong></td>
</tr>
<tr>
<td>Class activities each week</td>
</tr>
<tr>
<td>How parts of the class relate to each other</td>
</tr>
<tr>
<td>The grading system for the class</td>
</tr>
<tr>
<td>The self-paced structure</td>
</tr>
</tbody>
</table>

*Note: M=mean, S.D.=standard deviation, NH=No help, LH=little help, MH=moderate help, H=much help, VH=very much help, NA=not applicable.*
A Hybrid and Novel Approach to Teaching Computer Programming in MIS Curriculum

Table 11. Student assessment programming skills gained

| Item                                | M  | S.D. | NAT | ALI | SW | ALO | AGL | NA 
|-------------------------------------|----|------|-----|-----|----|-----|-----|-----
| Solving problems                    | 3.15 | 1.11 | 10% | 14% | 38% | 27% | 10% | 0%  
| Creating flowcharts                | 3.51 | 1.02 | 4%  | 12% | 29% | 38% | 16% | 0%  
| Reading and understanding C++      | 3.51 | 1.13 | 3%  | 18% | 25% | 30% | 23% | 0%  
| Writing C++ programs               | 3.08 | 1.20 | 12% | 19% | 30% | 27% | 12% | 0%  
| Debugging software                 | 3.42 | 0.99 | 2%  | 16% | 34% | 33% | 14% | 0%  
| Testing software                   | 3.16 | 1.08 | 6%  | 21% | 35% | 28% | 10% | 1%  
| Working effectively with others    | 3.77 | 1.45 | 10% | 10% | 20% | 24% | 27% | 10% 
| Explaining how programs work       | 3.44 | 1.19 | 4%  | 20% | 31% | 20% | 24% | 1%  

Note. M=mean, S.D.=standard deviation, NAT=Not at all, ALI=A little, SW=Somewhat, ALO=A lot, AGL=A great deal, NA=not applicable.

Table 12. Student assessment learning transfer

| Item                                | M  | S.D. | NAT | ALI | SW | ALO | AGL | NA 
|-------------------------------------|----|------|-----|-----|----|-----|-----|-----
| Core concepts of programming       | 3.42 | 1.15 | 4%  | 14% | 28% | 30% | 23% | 1%  
| How to solve problems              | 3.57 | 1.14 | 4%  | 19% | 31% | 25% | 20% | 1%  

Note. M=mean, S.D.=standard deviation, NAT=Not at all, ALI=A little, SW=Somewhat, ALO=A lot, AGL=A great deal, NA=not applicable.

Table 13. Elluminate usage response statistics

| Item                                | M  | S.D. | SD  | MD  | N   | MA  | SA  | NA 
|-------------------------------------|----|------|-----|-----|-----|-----|-----|-----
| Using Elluminate was helpful to my learning | 4.17 | 1.40 | 6%  | 3%  | 25% | 25% | 19% | 22% 
| I wish the course had used Elluminate more     | 4.22 | 1.49 | 6%  | 8%  | 22% | 6%  | 39% | 19% 
| I had difficulty setting up Elluminate       | 2.61 | 1.84 | 42% | 17% | 14% | 11% | 0%  | 17% 
| Elluminate was similar to face-to-face        | 3.97 | 1.40 | 3%  | 14% | 19% | 31% | 14% | 19% 
| Most courses would benefit from Elluminate   | 4.36 | 1.31 | 6%  | 0%  | 19% | 22% | 33% | 19% 

Note. M=mean, S.D.=standard deviation, SD=strongly disagree, MD=mildly disagree, N=neutral, SA=mildly agree, and SA=strongly agree.
The course instructor added six additional items to the instrument in a section labeled ‘Elluminate usage’ prior to its administration to gather relevant information from students as to whether the technology added value to their learning experience. One item simply asked whether students had participated in a synchronous session, while the remaining five items used a 5-point scale (1=strongly disagree, 2=mildly disagree, 3=neutral, 4=mildly agree, and 5=strongly agree) with a not applicable option. The scale demonstrates an acceptable degree of internal consistency reliability at $\alpha = .84$.

Though the descriptive statistics shown in Table 13 only highlight the usage of the tool during one academic semester, the results are promising. Fifty-nine percent of the respondents indicated the tool was easy to set up, and 44% indicated it was helpful in learning. Forty-five percent of the respondents indicated the tool was similar to a face-to-face experience, and 55% indicated most courses would benefit from the tool’s use.

**Outcomes on Satisfaction and Performance**

Inferential statistical tests were only performed on measures perceived to be valid and reliable. This includes the assignment performance scores, level of satisfaction subscale, and final course grade.

**Overall Performance and Retention**

Table 14 reports the mean performance on each assignment and the overall grade point average for the students that responded to the survey (not including Assignment 7). It is important to note that students that did not complete the assignments received a zero grade, but were included in the mean averages for each assignment. The data did not exhibit normality for all performance measures; however, there were not any severe departures from normality and ANOVA is robust to violations of this assumption.

<table>
<thead>
<tr>
<th>Performance measure</th>
<th>N</th>
<th>M</th>
<th>S.D.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final course GPA</td>
<td>245</td>
<td>3.00</td>
<td>0.95</td>
</tr>
<tr>
<td>Assignment 1</td>
<td>250</td>
<td>95.81</td>
<td>14.48</td>
</tr>
<tr>
<td>Assignment 2</td>
<td>250</td>
<td>88.44</td>
<td>18.78</td>
</tr>
<tr>
<td>Assignment 3</td>
<td>250</td>
<td>87.61</td>
<td>23.00</td>
</tr>
<tr>
<td>Assignment 4</td>
<td>250</td>
<td>81.33</td>
<td>26.32</td>
</tr>
<tr>
<td>Assignment 5</td>
<td>250</td>
<td>65.37</td>
<td>43.79</td>
</tr>
<tr>
<td>Assignment 6</td>
<td>230</td>
<td>48.78</td>
<td>47.68</td>
</tr>
</tbody>
</table>

From the fall of 2003-04 to the spring of 2004-05, the course averaged a 58% retention rate with 23% of the students withdrawing from the course and approximately 19% receiving a failing grade (D or F). The course retention rates improved from approximately 50% the first semester to approximately 65% in the last semester included in the analysis. Of the students passing the course, approximately 25% of the students earned an A, 22% earned a B, and the remainder earned a C grade.

**Gender**

To detect statistically significant differences on performance and satisfaction between males and females, one-way ANOVAs were constructed for numeric grades on each assignment, the level of satisfaction subscale, and the final grade in the course with gender serving as a between-subject condition. Results were statistically insignificant on every account as shown in Table 15.

Gender was also statistically insignificant when examining the satisfaction subscale $F(1,231)=1.09$, $p=.297$. Males reported a satisfaction scale of .681 ($SD=.131$) and females .705 ($SD=.124$). These are both positive findings in that they attest to the course not being characterized as gender bias on performance or student satisfaction. Both males and females appear to perform nearly the same and have a similar outlook on this course.
A Hybrid and Novel Approach to Teaching Computer Programming in MIS Curriculum

Programming Experience

To detect statistically significant differences on performance based on prior programming experience, one-way ANOVAs were constructed for numeric grades on each assignment, the level of satisfaction subscale, and the final grade in the course with prior programming experience as a between-subjects condition. Again, no statistically significant differences were detected as shown in Table 16.

Prior programming was also not statistically significant when examining the satisfaction subscale $F(1,231)=1.01, p=.317$. Those individuals with prior programming experience reported a satisfaction scale of .676 ($SD=.128$), while those without reported .703 ($SD=.127$). Again, both findings are positive in that they suggest the course content is appropriate for both novices and intermediate students. Both students performed nearly the same and were equally satisfied.

Employment Status

To detect statistically significant differences on performance and satisfaction among students employed full-time, part-time, or unemployed, one-way ANOVAs were constructed for numeric grades on each assignment, the level of satisfaction subscale, and the final grade in the course with employment-status serving as a between-subjects condition. This demographic resulted in a statistically significant difference on Assignment 2—$F(1,243)= 4.36, p=.01$—in favor of full-time employed students. All other measures were not significant, as shown in Table 17.

One might observe more variation in the final course grade based on employment status as it approaches significance $p=.07$. Unemployed students appear to achieve a higher final course grade than those students that are currently employed. This is a logical finding in that it suggests unemployed

### Table 15. Performance outcomes by gender

<table>
<thead>
<tr>
<th>Performance measure</th>
<th>Male</th>
<th>Female</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final course GPA</td>
<td>2.98</td>
<td>3.04</td>
<td>0.19</td>
<td>.66</td>
</tr>
<tr>
<td>Assignment 1</td>
<td>95.36</td>
<td>95.95</td>
<td>0.07</td>
<td>.79</td>
</tr>
<tr>
<td>Assignment 2</td>
<td>87.26</td>
<td>90.29</td>
<td>1.30</td>
<td>.26</td>
</tr>
<tr>
<td>Assignment 3</td>
<td>87.37</td>
<td>90.09</td>
<td>0.72</td>
<td>.40</td>
</tr>
<tr>
<td>Assignment 4</td>
<td>79.38</td>
<td>83.96</td>
<td>1.55</td>
<td>.21</td>
</tr>
<tr>
<td>Assignment 5</td>
<td>63.14</td>
<td>64.17</td>
<td>0.03</td>
<td>.87</td>
</tr>
<tr>
<td>Assignment 6</td>
<td>43.49</td>
<td>41.79</td>
<td>0.06</td>
<td>.81</td>
</tr>
</tbody>
</table>

### Table 16. Performance outcomes by prior programming experience

<table>
<thead>
<tr>
<th>Performance measure</th>
<th>Prior</th>
<th>No prior</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final course GPA</td>
<td>3.04</td>
<td>3.08</td>
<td>0.30</td>
<td>.59</td>
</tr>
<tr>
<td>Assignment 1</td>
<td>93.89</td>
<td>96.95</td>
<td>2.15</td>
<td>.14</td>
</tr>
<tr>
<td>Assignment 2</td>
<td>89.71</td>
<td>86.76</td>
<td>1.31</td>
<td>.25</td>
</tr>
<tr>
<td>Assignment 3</td>
<td>88.57</td>
<td>88.43</td>
<td>0.00</td>
<td>.96</td>
</tr>
<tr>
<td>Assignment 4</td>
<td>81.83</td>
<td>80.18</td>
<td>0.23</td>
<td>.64</td>
</tr>
<tr>
<td>Assignment 5</td>
<td>65.98</td>
<td>61.53</td>
<td>0.59</td>
<td>.44</td>
</tr>
<tr>
<td>Assignment 6</td>
<td>42.11</td>
<td>44.44</td>
<td>0.13</td>
<td>.72</td>
</tr>
</tbody>
</table>

### Table 17. Performance outcomes by employment status

<table>
<thead>
<tr>
<th>Performance measure</th>
<th>Unemployed</th>
<th>Part-time</th>
<th>Full-time</th>
<th>F</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final course GPA</td>
<td>3.19</td>
<td>3.03</td>
<td>2.82</td>
<td>2.72</td>
<td>.07</td>
</tr>
<tr>
<td>Assignment 1</td>
<td>95.49</td>
<td>95.76</td>
<td>95.79</td>
<td>0.01</td>
<td>.99</td>
</tr>
<tr>
<td>Assignment 2</td>
<td>88.15</td>
<td>84.87</td>
<td>92.95</td>
<td>4.36</td>
<td>.01</td>
</tr>
<tr>
<td>Assignment 3</td>
<td>86.22</td>
<td>87.84</td>
<td>87.79</td>
<td>0.11</td>
<td>.90</td>
</tr>
<tr>
<td>Assignment 4</td>
<td>81.86</td>
<td>80.60</td>
<td>81.51</td>
<td>0.05</td>
<td>.95</td>
</tr>
<tr>
<td>Assignment 5</td>
<td>61.17</td>
<td>70.66</td>
<td>63.35</td>
<td>1.07</td>
<td>.34</td>
</tr>
<tr>
<td>Assignment 6</td>
<td>47.85</td>
<td>52.51</td>
<td>45.99</td>
<td>0.42</td>
<td>.66</td>
</tr>
</tbody>
</table>
students, perhaps, have more time to devote to their academics. Employment status was not statistically significant when examining the satisfaction subscale $F(1,231)=0.07, p=.93$, indicating both employed and unemployed students had a similar degree of satisfaction with the course.

**Computer Programming Interest**

The final course grade and satisfaction subscales were correlated with the degree of a student’s interest in computer programming. There is a statistically significant, positive correlation ($r=.18, p=.004$) between the degree of interest in computer programming and overall performance and satisfaction ($r=.16, p=.014$) in the course. This is an expected outcome in that one would anticipate those individuals more interested in the content domain would achieve better performance and be more satisfied with the course.

**Satisfaction and Performance**

The final course grade and the satisfaction subscale mildly correlated ($r=.15, p=0.05$). Although statistically insignificant, the low correlation between the instrument and final course grade was somewhat surprising. One explanation could be a result of expectations. At the beginning of the Spring and Fall 2005-06 semesters, students were surveyed regarding their targeted course grades. The results were that 43% were targeting As, 26% were targeting Bs, and 31% were targeting Cs. Since the instrument was administered at the end of the academic semester, at a time when most students had a good estimate of their final grade, it is plausible that satisfaction was related to performance relative to target grade, rather than relative to actual grade. The results also provide strong evidence that students were willing to provide critical feedback about the course that was independent of their grade.

**DISCUSSION**

What can be concluded from this course and analysis, and what will be contributed to the best practices for computer programming instruction? The chapter has illustrated a number of challenges facing MIS computer programming instruction, and two key opportunities to impact MIS research and practice: the integration of ICT for instructional purposes, and the development, use, and validation of instruments designed to monitor our courses.

Of course, the results of this analysis must be interpreted in light of the limitations. This analysis has been conducted using data that were collected during different academic semesters. Technology, students, and curriculum has changed over the last 3 years, thus the design of the instrument has changed to collect relevant information. Movement of survey items within the survey, and modification or addition of items may have changed the constructs used in the analysis. In addition, combining responses from survey items to make composite variables may not adequately measure the constructs.

The degree of accuracy of these measures may also be questionable since the items are self-reported measures. Students, though informed their responses would have no bearing on their final grades, may have feared retaliation, and consequently, responded more favorably to the instructional methods. Further, since the instrument is completed by students at the end of an academic semester, experiences regarding assignments and methods earlier in the semester may be skewed. Though reliability and validity evidence could not be provided for all aspects of the instrument and some scales demonstrated less than acceptable internal consistency reliability, the authors wish to emphasize that instrument design is an on-going process.

In light of these limitations, this chapter has demonstrated the value of instruments to monitor our courses. Because university-wide course
evaluations only summarize a course at large, it is often difficult for educators to specifically monitor assignments and activities at a level of granularity that will inform instruction. The instrument demonstrated a level of granularity (assignment) that was useful for informing instruction. The chapter has attempted to demonstrate the validity of scores to measure students’ level of satisfaction with an MIS computer programming course, and in doing so, provided statistical conclusions demonstrating gender, prior programming experience, and employment status were not statistically different on student satisfaction in this course.

The instrument also provided some interesting descriptive results. Overall descriptive statistics indicate a positive outlook on learning computer programming skills and the transfer of these skills to different settings. Teamwork appears to be a favorable characteristic, from a student perspective, in computer programming instruction. And the use of synchronous communication tools shows great promise for computer programming instruction. This chapter has also described a novel and hybrid approach for the delivery of computer programming instruction in MIS curriculum using a variety of ICTs and pedagogical strategies. Some unique characteristics of the course include an assignment-centric design, self-paced completion, and the use of multimedia resources to replace lectures and provide flexible delivery of instruction. Analysis of student performance showed no differences in gender or prior programming experience, though a student’s employment status did demonstrate some variation.

In closing, the authors believe MIS faculty and administration should be mindful of a student’s perspective of their courses, and should take careful steps in the integration of novel ICTs and pedagogical strategies. We also feel that educators should not solely rely on university-wide course evaluations to efficiently and effectively capture this critical information from students. More relevant information can be collected by developing instruments tailored closely to our courses to aid in the decision-making process.

FUTURE RESEARCH DIRECTIONS

This research has documented the value of developing, validating, and using survey instruments to monitor and evaluate courses, demonstrated the use of novel pedagogical strategies (e.g., assignment-centric design), and provided many different forms of ICTs that can be gracefully integrated into the curriculum. While this chapter has provided evidence to demonstrate the instructional value of the technology and method, more work needs to be executed in this area. For instance, replicating the instructional methods and use of technology in hybrid classroom environments, both inside and outside of computer programming, is necessary to generalize findings and develop best practices to inform practice.

Educational research in computing- and information-centric disciplines also needs a stronger connection to other areas of educational research, an interdisciplinary approach. Of particular interest is the discipline of instructional technology. In recent years, researchers in instructional technology-related areas have been criticized for holding a one-directional view of the connection between theory and practice in which basic research questions or theory precede and gives rise to investigations having an applied focus or practice. The concept of developmental research or “design experiments” offers an exciting and useful alternative, whereby practical instructional interventions are rigorously studied for their usefulness in solving authentic problems (Reeves, 2000). Educational research without a strong grounding in a context and real-world connection, such as MIS curricula, lacks the necessity to inform practice.

Research in the realm of MIS curricula is open-ended with many opportunities for improvement and innovation, especially in computer programming instruction. Future research efforts should embrace an interdisciplinary approach, and include contexts to solve authentic instructional problems. The additional readings section includes
several readings that can serve as a starting place for this suggested and needed future research.

REFERENCES


Pennsylvania State University, American Center for the Study of Distance Education.


**ADDITIONAL READING**


A Hybrid and Novel Approach to Teaching Computer Programming in MIS Curriculum

dual-coding hypothesis. *Journal of Educational Psychology, 83*, 484-490.


