

AC 2009-2184: NEW APPROACH TO TEACHING AN INTRODUCTORY COMPUTER SCIENCE COURSE

Asad Azemi, Pennsylvania State University

Asad Azemi is an associate professor of Engineering at Penn State University. He has received his B.S. degree from UCLA in 1982, M.S. degree from Loyola Marymount University in 1985, and Ph.D. degree from University of Arkansas in 1991. His professional interests are in nonlinear stochastic systems, signal estimation, decision making under uncertainty, biocomputing, and use of computers and related technologies in undergraduate and graduate education to improve and enhance teaching and learning.

Nannette D'Imperio, Pennsylvania State University

Nannette D'Imperio is an instructor of computer science at Penn State Brandywine. She earned a B.S. degree in Elementary Education with a Mathematics Concentration from West Chester University and a M.S. degree in Computer Education from Philadelphia University. She has professional experience in customer service, departmental supervision, application support, and technical analysis. Her research interests lie in the use of programming languages, web-based learning and its applications.

New Approach to Teaching an Introductory Computer Science Course

Abstract

In this work, we present a new teaching approach that we have implemented in our introductory computer science programming course. The methodology consists of team teaching, a hybrid delivery system, recorded lecture retrieval capability, readiness assessment activities, objective assessment of student progress, and cooperative learning through team work. The team teaching approach consists of two faculty members being present and actively involved in lecture delivery and classroom activities, which take place in a computer laboratory setting. The hybrid delivery system consists of using Centra, a real-time communication, collaboration and learning software environment, for lecture delivery, recording, and active student participation. The recorded lectures are available to students for downloading and viewing at any time using any computer. The readiness assessment activities are incorporated to enhance student preparation for classroom lectures and are assigned on a weekly basis. The goal of the objective assessment part is to provide individual feedback to each student regarding mastery of ongoing course objectives through collective evaluations of quizzes and homework submissions. This process is repeated on a monthly basis during the semester. Cooperative learning is accomplished by forming student teams who work together on homework assignments and classroom activities. Details of this work, including its advantages, disadvantages, and student feedback, will be included in the paper.

Introduction

The usual approach to teaching a computer programming course is to have a classroom lecture component and a supervised laboratory module, where students write simple programs reflective of the subjects that they have learned during the lectures. For smaller class sizes, an alternative approach, which consists of conducting the entire course in a laboratory setting, has also been implemented. An improvement to this approach involves making the lectures more interactive, transforming them using a multimedia package, and conducting class sessions in a "technology-ready classroom" ¹. This requires a larger investment, and more time and money, on the parts of the institutions and instructors. Justification for this type of approach is apparent when we consider the basic problems of teaching an introductory computer programming course, which include the following. (i) Although the incoming students have considerably more experience with computers than their predecessors did, their experiences for the most part are limited to those of an "end user," and not a "developer." Therefore, the concept of programming is foreign to many of them. (ii) It is well understood that a programming language cannot be learned by just reading a textbook or listening to class lectures. One needs to read the textbook, attend lectures, and more importantly, practice the subject matter by writing programs. This is the justification

for adding laboratory modules to programming courses. The subject material is reinforced with the inclusion of hands-on programming exercises shortly after the presentation of new concepts, which promotes active learning²⁻³. Another enhancement, which has been employed, is the use of the problem-based learning approach⁴⁻⁵. In this work, we describe our new approach to teaching an introductory programming course that consists of team teaching, the incorporation of a hybrid delivery system, the utilization of a variety of assessment tools, and the employment of cooperative learning through team work. Although some of these approaches have been implemented individually by other researchers and the authors, the combination of the aforementioned tools has not been tried as a single methodology. The use of objective assessment, which is mainly a program assessment tool, has not been employed in the evaluation of student progress. To the best of our knowledge, the utilization of readiness assessments, to help students gain a better understanding of the topics that will be covered in class, has not been implemented in the delivery of introductory programming courses.

Course Structure

The course material resembles a typical introductory C++ programming course for Computer Science, Computer Engineering and Information Science and Technology students, including additional topics in structures, such as arrays of structures, working with external files employing structures, and some exposure to dynamic arrays. The course grade is a weighted average calculated based on scores earned on weekly readiness assessments (15%), weekly homework assignments (20%), weekly quizzes (20%), and two exams (45%). The more challenging quizzes are allotted more time and administered as online timed take-home problems. The quiz grade is calculated after dropping the two lowest scores. Exams consist of two parts: (i) an objective portion in the form of true-false and multiple choice questions, and (ii) a programming portion. These parts are administered on different days. All programming submissions are graded using a published “program check list,” which describes all required components and associated point values to achieve an assignment grade (see Figure 1). All course materials, including quizzes and exams, are administered and accepted using the web. Paper submission is not an acceptable option.

Proposed Methodology

Team Teaching

The team teaching approach consists of two faculty members being present and actively involved in lecture delivery and classroom activities. The lead instructor delivers the lecture material using the Centra delivery system. The second instructor is positioned primarily in the back of the classroom to monitor student computer involvement and assist students with technical difficulties. We have found this to be more productive than using classroom management software, e.g. GenevaLogic Vision software⁶, to monitor and control student activities. This

approach allows the lead instructor to focus more on lecture delivery while the second instructor monitors student involvement in the class material.

During in-class exercises, both instructors are available to assist and guide students, while they work on team programming activities. In the case of hands-on coding demonstration, one instructor guides the students through the process, while the other instructor assists those having difficulties.

We recommend that course related tasks are divided between the instructors based on individual preferences. In our case, lecture materials, quizzes and exams were prepared jointly and grading was divided evenly. At the end of each class, additional help was provided primarily by one instructor.

The team teaching approach was adopted since we were teaching two separate sections of the same course. Due to the lack of a programming placement test, at our university, these sections typically consist of a diverse student population, ranging from a strong programming background to little or no experience. Our goal with this approach was to provide students with more individual attention and assistance, while maintaining the pace of the course. We should mention that in cases that graduate teaching assistants are available this approach could be accomplished by having a faculty and 1 or 2 assistants.

Hybrid Delivery System

The usefulness of hybrid delivery in education has long been realized and with the advancement of computer and communication technologies and the introduction of Web based authoring tools, its effectiveness has been further extended⁷⁻⁸. Our hybrid approach consists of several components: (i) lecture delivery in a computer laboratory setting using an online synchronous delivery system, which allows classroom and/or Internet attendance; (ii) asynchronous delivery of classroom lectures via downloadable recordings; and (iii) use of a course management system.

Centra⁹, a real-time communication, collaboration and learning software environment, is employed for lecture delivery, recording, and facilitating active student participation. The instructor can share the computer screen, document, or an application. Typical PowerPoint lecture presentations are converted into individual GIF, JPEG or HTML slides/pages during the upload process. The instructor presents the material one screen at a time and can jump to any screen by just clicking the slide title. There are many available features, which can be used to enhance the presentation. The built-in “laser pointer” and “highlighter” capabilities are useful for stressing important points and topics on the slides. The whiteboard feature, much like a physical white or chalk board, allows the instructor to create drawings, diagrams, and free typing and/or writing to further explain the concepts being presented in the slides.

There are also interactive functions that are helpful in soliciting responses from students. For example, the instructor can poll a multiple choice question and each student chooses one answer.

The answers from the students are available to the instructor anonymously in real-time. Centra will also allow students to download the recorded sessions for later viewing at their own convenience. Based on our experience, we consider this a significant advantage. Figures 2 and 3 provide sample screen shots.

ANGEL¹⁰ (A New Global Environment for Learning) is the University's CMS (Course Management System) tool that is used for posting courseware modules, the course syllabus, the daily schedule, quizzes, exams, readiness assessments, homework folders, attendance, and grading. ANGEL offers a timing capability to control access that is used extensively for all time-dependent assignments, quizzes and exams.

Readiness Assessment

The primary objective of each readiness assessment is to make sure the students gain a general understanding of the subjects that will be covered in class, attain more information from the presentation, and become more engaged in classroom activities.

Each questionnaire consists of a number of true/false and multiple choice questions that can be answered in a 20-30 minute time frame (see Figure 4). ANGEL's quiz feature is used to randomly generate a 15-20 question quiz from a bank of preloaded problems. Students are allowed multiple attempts (7-10 depending on the subject difficulty) to achieve a minimum score of 70-75%. ANGEL retains all of the quiz attempts and, by our design, records only the highest score in the course grade book. Students who achieve a score of 70-75% are given a grade of 100% for that assessment. Any score below the minimum is graded in proportion to that achievement. For example, a score of 60% out of a 75% minimum requirement would receive a grade of 80%. These grades are entered manually in ANGEL.

Objective Assessment

The goal of the objective assessment part is to provide individual feedback to each student regarding mastery of ongoing course objectives through collective evaluations of quizzes, readiness assessments, exams, and homework submissions. This process is repeated on a monthly basis during the semester. The objective assessment feedback is based on breaking down each chapter's goals into more specific measurable outcomes. Each outcome is rated using a scale of 1 to 5, where 1 indicates that the goal is not met and 5 indicates that the objective is satisfied. This process is used to help students identify areas that they need to improve based on their performance in various class assignments. In a way, it summarizes their activities and provides them with specific feedback by topic. For example, we evaluated the topic "C++ Fundamentals," which had several outcomes, such as:

- Be able to correctly use different data types and arithmetic operators.
- Be able to use assignment statements, format floating numbers, and use math libraries.

- Be able to write functions without arguments.

Based on his/her performance, students were assigned a scaled value for each outcome. Figures 5-6 provide two assessment samples. We should point out that the role of objective assessment is to help indicate mastery of subject matter as well as areas of weakness. The objective assessments can be considered as qualitative translations of a student's performance based on his/her graded assignments. Moreover, it helps the instructor evaluate achievement of his/her teaching objectives and, if a certain area indicates weakness by a majority of students, the instructor needs to spend more time or change his/her approach for that topic.

In the majority of cases, students are unable to identify the intended relationships between the class assignments and the intended learning objectives of the course. The objective assessment can fill this gap.

Cooperative Learning through Team Work

Cooperative learning is accomplished by forming student teams who work together on homework assignments and classroom activities. Teams of three were originally formed based on seating. After the first objective assessment, teams were restructured, by the faculty, so that each would include at least one strong student, identified by prior graded work.

During the in-class programming activities, students were asked to work with their teammates and come up with a solution to the assigned problem. As others have observed, there are always dysfunctional teams that need some special attention and, possibly, change of membership.

Students were allowed to work with their teammates and submit identical homework assignments, as long as all names were included as contributors in the program code and the submitted sample runs were their own. Realizing that this could result in copied submissions by weaker students, we encouraged students to learn from one another, contribute to the final result, and only submit code that they fully understood. For the most part, they realized the value of this approach and honored this arrangement.

We also promoted the value of teamwork, in general, to imitate the real world condition.

A Typical Class Meeting

Students are expected to take a readiness assessment questionnaire before the first lecture for that week. Prior to each class, session material, in PowerPoint format, is uploaded to a scheduled event on the Centra delivery system. This includes the daily agenda, required weekly readings, assigned homework problems, deadlines, and the session lecture material. Upon entering the classroom, students log into the Centra system and attend the event for the day. Students, who are not physically on campus, may log in from anywhere using a computer with Internet access and speakers and/or a headset to attend the lecture in real time. Once the instructor starts the Centra session, each student will be able to view each slide as it is being presented, by the

instructor, on his/her own computer screen. The lecture material consists of several segments of 10-15 minutes of instruction followed by an in-class exercise or activity. Classroom instruction consists of presenting material using imported PowerPoint slides in Centra, switching to the Microsoft Visual Studio environment for program illustration, compilation and execution, illustrating concepts and logic via the Centra whiteboard, and sharing code segments with the entire class by using Centra's text chat feature. The Centra system also provides the instructor the capability of sharing a student's work with the entire class and the ability to interactively modify and correct his/her program. When they have a correct answer, we randomly choose a team's or a student's work and share the solution, through Centra, with the rest of the class. In case of a problem or coding question, the program is shared, through Centra, so that suggestions or corrections can be made by the class. This feature was used extensively throughout the semester. All of these activities are recorded by Centra and available for student downloading and viewing.

The instructor can view a list of all attendees on the Centra screen at all times during the event. Attendance time is logged and stored for later viewing via the Centra reporting mechanism.

Remarks

The following items summarize our observations:

- The time and effort needed to plan, develop, and deliver the course using this approach, for the first time, was much more than delivering the course with previous methodologies as individual instructors.
- The process requires several test runs in order for faculty to get comfortable with the synchronous delivery system.
- Our experience indicates that it will take students 2-3 weeks to get comfortable using the synchronous delivery system and connecting from outside the campus.
- Class participation did not change much, but the ability to share student work with the rest of the class has had a positive effect.
- For those students who did not fully participate in the learning opportunities available during and after class, homework grades did not improve. On the other hand, those who engaged in these activities showed improvement in their homework grades, compared to previous years.
- Student reactions, based on an online survey, have been positive and encouraging.
- The ability to see the slides and presentation on their own screens and to interact with the instructor through the Centra system was referred to as the most positive aspects of the

new delivery system. After that the ability to download or connect from outside the campus was cited.

- Readiness assessment quizzes play an important role in student preparation and participation in class discussions.
- The primary data based on number A's in the course suggests, that the process that we implemented, helped the more motivated students to achieve higher grades. The percentage of A's increased from a 20-25% range to 40%. As of yet, we do not have adequate data to make the same type of observation of the less motivated students.
- Student response to the adopted approach was overwhelmingly positive, even from the struggling students.
- We believe that the presented methodology is scalable to handle larger class sizes. The aforementioned benefits are potentially greater.

The following items summarize our recommendations:

- Include post-assessment components to evaluate if the learning objectives have been met and identify any areas of weakness. We recommend using the same question banks as the readiness assessments for this process.
- We believe that team teaching at a smaller university campus can replace the lecture-recitation approach that is used in larger institutions. Moreover, since the students had the opportunity to stay in the classroom for optional assistance, the learning was not postponed and we believe this is very effective for those who take advantage of this benefit.
- A cost benefit analysis versus other approaches would be useful.
- Cooperative learning portion could be further emphasized and encouraged by incorporating it in the course grading scheme.

We plan to include more data gathering the next time that we deliver this course so that we can better assess the strengths and weaknesses of this approach.

Conclusions

In this work, we have presented an approach that can make class lectures more informative and enjoyable for the students. The methodology consisted of team teaching, a hybrid delivery system, recorded lecture retrieval capability, readiness assessment activities, objective assessment of student progress, and cooperative learning through team work. The time requirement for producing lecture modules and developing the various assessment activities was

substantial. However, based on our initial data, we believe the value achieved justifies the extra effort.

References

1. Azemi, A., "Teaching Computer Programming Courses (Using the Internet) in a Computer Laboratory Environment," *Proceedings of the 2002 ASEE Annual Conference and Exposition*.
2. Luo, J, and R. Hovis, "Promoting Active Learning in Teaching the Organization of Programming Languages Course," *Proceedings of the 2006 ASEE Annual Conference and Exposition*.
3. Barrière, C., "Using the Web to Promote Active Learning Outside of Class Time," *Proceedings of the 2002 ASEE Annual Conference and Exposition*.
4. Said, H. and F. Khan, "Towards Using Problem-Based Learning in Teaching Computer Programming – Step 1: Developing Synchronized Multimedia Lectures Using Video and PowerPoint," *Proceedings of the 2004 ASEE Annual Conference and Exposition*.
5. Duch, B., S. Groh, and D. Allen, *The Power of Problem-Based Learning*, Stylus Publishing, VA, 2001.
6. GenevaLogic Company, Beaverton, OR 97006.
7. Young, J.R., "Hybrid teaching seeks to end the divide between traditional and online instruction," *Chronicle of Higher Education*, vol. 48, 2002, pp. A33–A34.
8. Azemi, A., "Enhancement of Traditional and Distance Learning through Hybrid e-learning Approach," *Proceedings of the 2008 ASEE Annual Conference and Exposition*.
9. Saba Software Inc., Redwood Shores, CA 94065.
10. ANGEL Learning, Indianapolis, IN 46278

1.	Program Execution	
	a. Program runs and gives all correct answers.	50
	b. Program runs and gives at least 1 correct answer.	30
	c. Program runs but gives no correct output.	20
2.	Program documentation: Program Synopsis	
	a. Synopsis has few or no corrections by the instructor.	5
	b. Synopsis has comments but is an adequate summary of the program.	3
	c. Inadequate synopsis.	0
3.	Program Documentation: Comments	
	a. Program is well commented	10
	b. Program has comments.	5
	c. Inadequate commenting.	0
4.	Program Documentation: Programming Style	
	a. Variable names and procedure names are meaningful. There is good use of "white space," indentation and other formatting items. Program will be understood by the unfamiliar reader.	10
	b. Program will be understood by an experienced reader.	5
	c. Inadequate style.	0
5.	Output Format	
	a. Output satisfies all requirements of the assignment.	10
	b. Output satisfies some requirements of the assignment.	5
	c. Inadequate.	0
6.	Test Data	
	a. Program has exhaustive or extensive set of test data.	15
	b. Program tests for reasonable and obvious values.	10
	c. Inadequate test data.	0

Figure 1. Program Check List

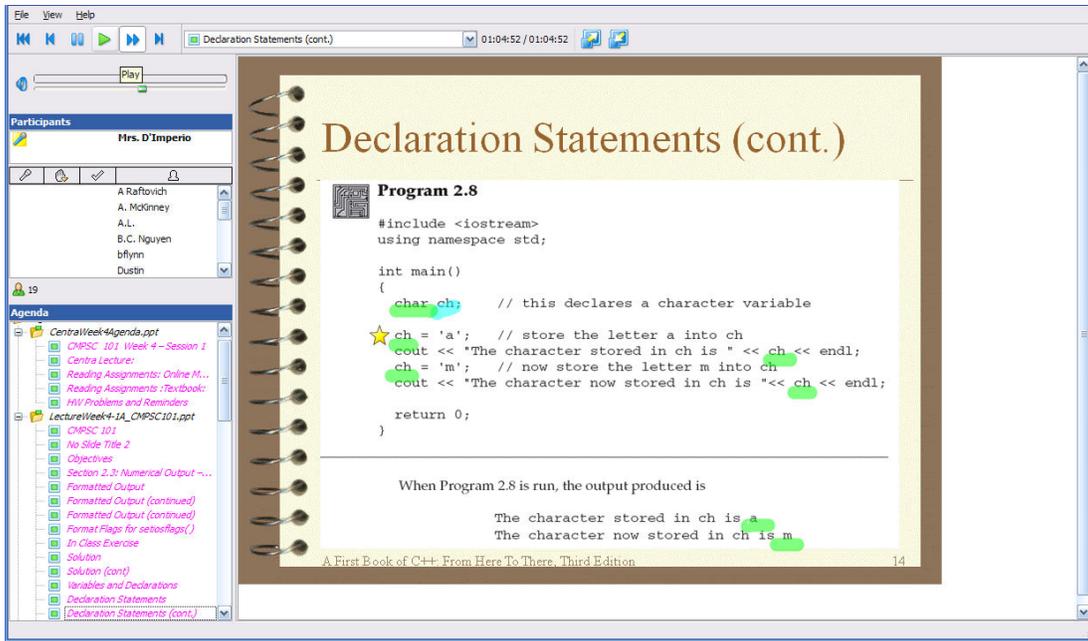


Figure 2. A screen shot from Centra recordings.

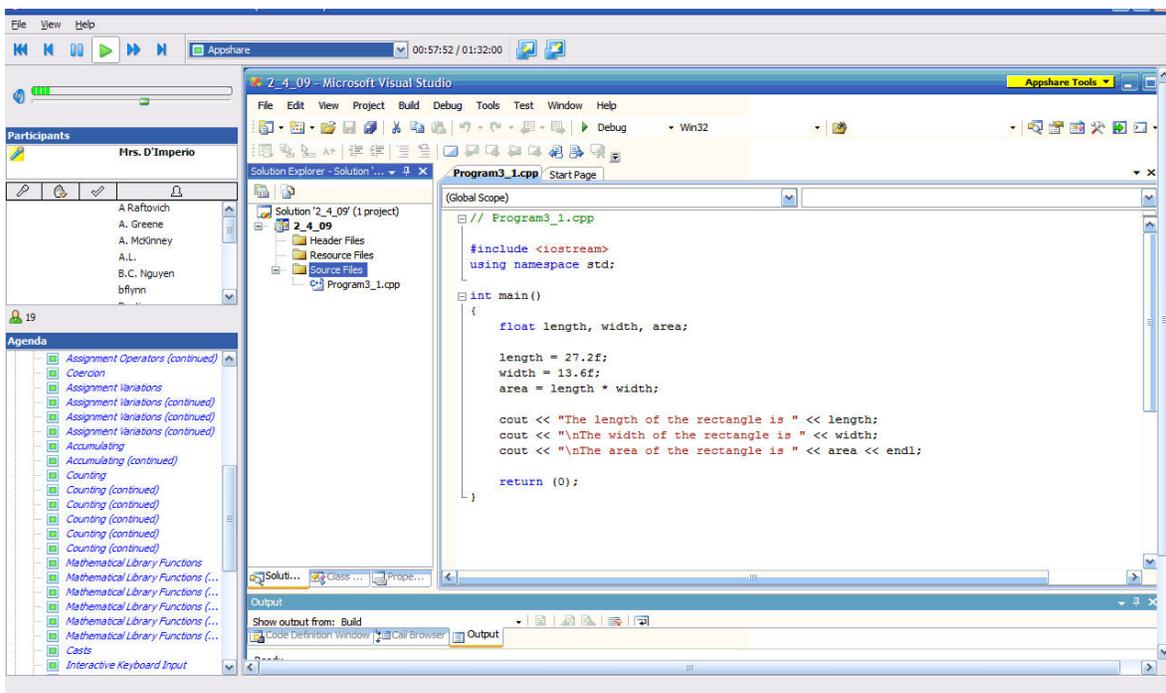


Figure 3. A screen shot from Centra recordings, showing an AppShare.

Readiness Assessment: Prep for Week 11

1. The indexed variables (members) of an array must be integers.

- A. FALSE
 B. TRUE

2. Arrays can be passed to functions.

- A. FALSE
 B. TRUE

3. The formal array parameter in a function represents the address of the first element in the actual array argument.

- A. FALSE
 B. TRUE

4. An array is always passed by reference.

- A. FALSE
 B. TRUE

5. The following function definition has an error in it. What line is this error on?

```
0. void f1(const double array[], int size)
1. {
2.     int i=0;
3.     while(i< size)
4.     {
5.         array[i] += 2;
6.         cout <<array[i];
7.         i++;
8.     }
9. }
```

- A. 0
 B. 2
 C. 5
 D. 6

9. Which of the following will read values from the keyboard into the array? (Assume the size of the array is SIZE).

- A. cin >> array;
 B. cin >> array[];
 C. cin >> array[SIZE];
 D. for (i = 0; i < SIZE; i++)
 cin >> array[i];

Figure 4. Partial readiness assessment questionnaire

CMPS 121 Assessment

Scaling: 1 (objective not met) → 5 (objective satisfied)

Student 1

Topic	History of computing and programming languages
Outcome	Be able to identify key development in the history of computing and programming languages.
Assessment	4
Topic	Overview of C++
Outcome	Be able to use Visual C++.net to write, compile and run a program. Write a simple C++ that can output a static message/calculation.
Assessment	5
Topic	Overview of C++
Outcome	Understand data types and variable declaration in C++. Be able to perform simple dynamic data entry.
Assessment	5
Topic	C++ Fundamentals
Outcome	Be able to correctly use different data types and arithmetic operators
Assessment	5
Topic	C++ Fundamentals
Outcome	<i>Be able to use assignment statements, format floating numbers, use math libraries</i>
Assessment	5
Topic	C++ Fundamentals
Outcome	<i>Be able to write functions without arguments</i>
Assessment	5

Figure 5. Sample Objective Assessment

CMPS 121 Assessment

Scaling: 1 (objective not met) → 5 (objective satisfied)

Student 2

Topic	Working with Functions
Outcome	<i>Be able to write functions with arguments</i>
Assessment	4.5
Topic	Selection Process
Outcome	<i>Be able to use selection structure statements to solve problems with multiple alternatives</i>
Assessment	3.5
Topic	Repetition and Loop Statements
Outcome	<i>Be able to apply loop statements to solve problems that involves repetition</i>
Assessment	4
Topic	Modular Programming
Outcome	<i>Be able to write functions with value and reference parameters</i>
Assessment	3.5
Topic	File input-output (1)
Outcome	<i>Be able to write to a file and read from a file, without a function</i>
Assessment	4
Topic	File input-output (2)
Outcome	<i>Be able to write to a file and read from a file, using a function</i>
Assessment	4

Figure 6. Sample Objective Assessment