Combining Active Learning Approaches for Improving Computing Course Outcomes at Minority-Majority Institutions

Dr. Debra Lee Davis, Florida International University, School of Computing and Information Sciences

Dr. Debra Davis is an Instructor in the School of Computing and Information Sciences at Florida International University. Her research interests emphasize interdisciplinary topics including understanding and improving: (1) Computer Science education, including increasing participation of women; (2) educational applications and techniques for online STEM learning; and (3) complex human-machine interactions. She has a Ph.D. and M.A., in Cognitive Developmental Psychology from the University of Texas at Austin, and an M.S. in Computer Science from FIU.
Combining Active Learning Approaches for Improving Computing Course Outcomes at Minority-Majority Institutions

Abstract
Research shows that over 50% students who try an introductory course in programming do not complete the course. At Hispanic-serving institutions (HSI) and those with large populations of 1st generation college students, this is particularly problematic. We combined and customized approaches for use in an introductory programming course for Computer Science Majors at a large HSI. The approaches included the use of: (1) Collaborative, team-based and paired programming, active learning, in-class exercises, as well as additional external assignments; (2) Active learning classroom environment whereby the physical space enhances and encourages collaborative, small group interactions; and (3) In-class Peer Learning Assistants (undergraduates) that have undergone specialized training to facilitate discussion and interaction with students in an active learning classroom setting. We conducted a study in a Programming I for Computer Science Majors (CS1) course to test the efficacy of the 3-pronged approach described above. The control group (lecture based) pass rates were found to be 71%, whereas the experimental group (active learning) pass rates were found to be 80%. For those students who passed the class, average grades also increased, with the average grade in the control group at 3.0 and the average grade in the experimental group at 3.30. It is thus concluded that the use of the active learning and the 3-pronged approach described above had a positive impact on course outcomes and student learning.

1. Introduction
In recent years, there has been a dramatic and rapid increase in enrollment in computing courses. This has put increasing pressure on Computer Science Departments to provide greater access to, and success in, key computing courses. For universities that primarily serve underrepresented minorities and first generation college students, the challenges are even greater. Our work seeks to address this by combining and testing several active learning approaches in fundamental computing courses at a minority-serving university.

Focus on Diversity: Florida International University (FIU) is a public university located in Miami, Florida. As a Hispanic-Serving Institution (HSI), FIU is committed to successfully educating minorities and first generation college students. FIU’s commitment to this can be seen in its many successes.

- FIU graduates the most Hispanic computing and engineering majors in the continental US
- Nearly 80% of FIU’s students are minorities, and many are first generation students
- FIU’s School of Computing and Information Sciences (SCIS) graduates the 7th largest number of computing majors in the US, per ASEE;
- FIU’s Information-Technology programs were recently ranked #1 in the state by the Florida Board of Governors, ahead of more recognized universities such as UF and FSU
2. Problem Statement

Research shows that over 50% students who try an introductory course in programming do not complete the course. Those that stay oftentimes fail to meet the requirements to pass the course [1]. At HSIs and those with large populations of 1st generation college students, such as FIU, this is particularly problematic. The failure/drop rate for FIU’s two introductory programming courses is over 50%. Other required courses that entail strong programming and math skills, such as data structures, have drop and failure rates close to 50%. Finding solutions to improve student retention and performance in these courses has been a major challenge. New and innovative approaches are greatly needed.

3. Prior Work

Prior research has found that active learning approaches can improve student learning outcomes and retention in STEM fields. For example, peer-led guidance has been found to positively impact passing rates, and decrease course re-take rates in physics and math classes [2]. Specifically, peer-led team learning tends to [3]:

- Facilitate active learning in introductory STEM courses
- Promote a high level of engagement by students
- Increase class attendance rates
- Increase student performance by at least 15% in overall grades
- Increase success of women and minorities

Flipped classroom approaches have found success for improving outcomes [4,5,6,7] in the following areas:

- Provide an opportunity for students to gain a first exposure prior to class
- Provide an incentive for students to prepare for class
- Provide a mechanism to assess student understanding
- Provide in-class activities that focus on higher-level cognitive activities

Online, interactive tutorials have been found to help improve student performance in introductory computing courses by 16% overall. The impact was greatest for students in the lower quartile, whose improvement was found to be at 64%. Student engagement has also been found to improve when using these types of tools [8,9].

Most of this research has looked at these approaches in isolation, and have not addressed their impact at universities that primarily serve minorities and first generation college students. However, FIU’s Department of Physics conducted a series of studies that combined 3 approaches to transform introductory Physics labs: (1) peer-led guidance using the University of Colorado at Boulder’s Learning Assistant model; (2) cooperative group learning via modeling; and (3) the University of Maryland, College Park Online Curriculum. Prior research had found that at non-minority institutions, these combined approaches significantly improved student retention and learning outcomes with moderate to high impact, when compared to traditional approaches [2,3].

The Department of Physics at FIU conducted a series of studies that applied a number of those approaches and found that 3 of them improved student retention and learning in introductory physics courses [10,11]. These included:

- Peer-Led Learning Assistants in an Active Learning classroom
Combining Active Learning Approaches for Improving Computing Course Outcomes

- Collaborative, active learning (i.e., flipped) within the classroom
- Online, just-in-time, tutorials as an integrated part of the curriculum

In our approach, we expanded and customized these approaches for use in an introductory programming course for Computer Science Majors. By using this approach, we expected the following outcomes:

<table>
<thead>
<tr>
<th>Outcome</th>
<th>Measured by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greater retention and passing rates</td>
<td>Final grades, drop rates by course and semester</td>
</tr>
<tr>
<td>Improved performance in key computing concepts</td>
<td>Comparison pre/post CS Concept Inventory performance [12]</td>
</tr>
<tr>
<td>Higher student engagement, as indicated by greater student preparation for class</td>
<td>Quiz performance and online tutorial completion rates</td>
</tr>
</tbody>
</table>

4. Approach and Implementation

How our approach is different: The combined approach implemented in FIU’s introductory Physics labs had not yet been tested in CS/IT courses. Our approach incorporated the same 3 pronged approach used in the physics labs along with the following customizations for computing courses:

4.1 Peer-Led Learning Assistants (LAs)

Peer LAs are undergraduates who have previously successfully completed the course. During both class and lab, LAs circulate around the room, answering questions, and providing more individualized help for different groups of students. This allows students to work on these types of problems during a time when they have access to someone who can help them when needed. Because they are the students’ peers, they are often seen as less intimidating than the instructor, and are more likely to be approached by some students.

The typical ratio is 1 LA for every 20 students in a course. LAs undergo specialized training to facilitate discussion and interaction with students in an active learning classroom setting. This training includes a one semester course on active learning, pedagogy and associated approaches to use as an LA, as well as semester long training sessions with the instructor and senior LAs. During class, LAs facilitate collaboration among student teams and help students understand and solve problems, without directly providing the answers. Instead, LAs are trained to ask questions of students and provide guidance that can help them discover how to solve a problem themselves. They also engage with all students who are working in a group (see 4.2 below) to help the students actively work collaboratively on the in-class exercises.

In addition to in-class work, peer LAs hold regular office hours in the general CS computing labs on campus. Their schedules are posted to the course LMS page, and students can make regular appointments with them or meet with them as walk-ins. As part of their office hours, LAs will hold out of class review sessions prior to each exam. These sessions involve having students
engage in hands-on exercises using the types of problems they would be expected to have mastered for the particular exam.

LAs partner with the instructor by regularly providing valuable feedback about the level of understanding of the students they are facilitating during each class activity. This allows the instructor to change the teaching approach to better fit the learning needs of the students prior to any assessments such as quizzes or exams. Because there is more one-on-one interactions with individual students, it also provides the instructor with more individualized understanding of which students are struggling and may need additional intervention. Unlike TAs, they do not grade student work or help manage the classrooms.

4.2 Participatory, collaborative learning
Participatory, collaborative learning entailed several components:

- Collaborative, team-based and paired programming, active learning, scaffolded in-class exercises, as well as additional external assignments
- Active learning classroom environment whereby the physical space enhances and encourages collaborative, small group interactions

Importantly, this approach allows students to practice the theory taught in the course while in the classroom where they can easily get help from the instructor and peer LAs.

The active learning approaches employed in our approach included:

- In-class, scaffolded programming exercises where students were engaged in solving problems and exercises that began with simpler concepts and problem solving skills, and progressed to more complex exercises that built upon what was learned in the prior exercise. For example, one exercise would ask students to build a basic if statement using specified criteria (e.g., create an if statement that prints ‘go’ if the variable light equals 1 and ‘stop’ if it equals 0). The subsequent exercise would ask the students to build a more complex example (e.g., a nested if statement), and the last exercise would ask students to build an if statement using a more abstract question (e.g., You have an elevator that goes to the following named floors, 12, 14 and 15, which actually correspond to the following physical floors, 12, 13, 14. Write an if statement that would ensure that users go to the correct physical floors given the button they push (named floor)).

- A collaborative, team-based approach was used in the lecture portion of the classroom for problem solving exercises. Groups were typically 4-5 students who worked on problems together. The instructor, TAs and LAs help guide students to work collaboratively with each other by engaging with and asking questions of all students in each group, and having the students actively engage with each other using similar approaches.

- All labs employed the use of pair programming whereby students would switch roles (driver who writes the code and navigator who evaluates it) after each problem so that each student would have equal opportunities to learn in each role.

The active learning classrooms used spaces with large tables that could accommodate up to 6 people. Chairs were movable and the physical space surrounding each table could easily be
traversed. The physical configuration of the room and tables encouraged a natural formation of groups of students. This is quite unlike traditional classrooms with individual rows of fixed or difficult to move desks, where students are unable to easily sit together as a group and there is very little room to walk around and work with groups of students.

4.3 Online, Just-in-time tutorials
Online, just-in-time tutorials provide students with real-time feedback to help students understand background knowledge prior to attending a class, so students will come to class ready to apply and practice their knowledge in guided, in-class activities. Instructors are able to monitor progression of each student and the class as a whole.

Examples of these online resources include: Introduction to Programming curriculum from Princeton University [13]; Zyante [14]; Problets [15]; and OpenDSA [16].

5. Research Design
We conducted a study in a Programming I for Computer Science Majors (CS1) course to test the efficacy of the 3-pronged approach described above. The study was conducted across two semesters, Spring 2015 and Fall 2015, and involved multiple sections each semester. The experimental design is discussed below.

5.1 Experimental Approach:
Groups - There were 2 groups that consisted of multiple sections from the same semester: Control group (Spring 2015) and Experimental Group (Fall 2015).

The Control group received:
- Standard, lecture-based presentations of the course materials
- Class was held in a traditional classroom setting (fixed seats set in rows)
- Active learning exercises consisted of approximately 15% of the time during in-class lectures, and
- An average of 3 student volunteers were present during class to support any questions the students might have while working on the class exercises

Importantly, unlike LAs, the student volunteers received no LA-related training, did not hold office hours and did not provide the instructor with feedback regarding the level of understanding of the students.

The Experimental group experienced the three-pronged approach discussed above, including:
- A scaffolded, active learning (i.e., flipped) classroom approach that included the use of LAs in both lecture and lab components
- Class was held in an active learning classroom where the physical space is designed to encourage small group interactions
- Active learning exercises consisted of approximately 85% of the time during in-class lectures
- Formally trained LAs were present during both class and labs to assist students who were having difficulty with the active learning exercises
• Each LA held regular office hours (in addition to Instructor and TA office hours) where they would meet with students to review course content, help with difficulties encountered with assignments, etc.

**Course Assignments and Exam** – Course assignments and exams were very carefully designed so that each assignment and each test question between the semesters were evaluating the same concepts and skill sets. This was not merely by general topic, but instead, addressed specific capabilities and requirements, assignment-by-assignment, and question-by-question, for each exam.

For example, questions on each exam were analyzed and revised question by question to ensure that the same concept (e.g., math functions) and level of knowledge (based on Bloom’s Taxonomy, e.g., both questions fall under ‘apply’ or both fall under ‘create’) were equivalent. Some questions on the exams were also not modified for even more direct comparisons.

**Pretest/Posttest Concept Inventory** - Each course section was administered pre- and post-test concept inventory at the beginning and end of each semester, respectively, to assess their knowledge of the course material. The pre/post tests consisted of a CS programming concept inventory [12].

**Active Learning & Collaborative Spaces** – The control group class meetings were in traditional, lecture-based classrooms. As part of the active learning approach, class meetings for the experimental group was conducted in active learning classrooms. These flexible classrooms can accommodate 60-173 students each, and are equipped with multiple projectors, screens, glass “white” boards, tables and rolling chairs, and other physical design elements that are conducive to students actively participating and collaborating with others in their learning process.

5.2 Research Challenges:

There are a number of challenges that must be addressed in our research approach.

• **Quasi-experimental Design** – Ideally, participants should be randomly assigned to either the control or experimental group. That, however, is not practical in our case. Instead, Spring 2015 courses were used as our control and Fall 2015 courses were used as our experimental group. Importantly, all participants were administered pretests, and comparisons were made between the sections to ensure that there were no significant differences between them in their initial knowledge and understanding of the course content.

• **Experimental Controls** – Strong experimental controls were used to ensure validity, including:
  
  o **Materials & Resources** - The students in both the control and experimental groups had access to the exact same course materials and resources. Assignments, labs and exams were carefully designed to assess the same specific concepts and skills. The pre/post concept inventory used was also the same for each group.

  o **Instructors** – Different instructors often teach different sections of the same course, introducing a potential confounding variable. In our study, the same Instructor taught all sections in both the control and experimental groups, thus eliminating this issue.

6. Results

The control group consisted of a total of 85 students in one section. The experimental group consisted of a total of 171 students across two sections, with one section containing 43 students
and the second lecture containing 128 students. To determine whether class size might have an impact on student outcomes, pretest/posttest and exam score results were compared across the two experimental sections. No significant differences were found between the two experimental sections. As a result, the two sections were analyzed as one experimental group.

Results found a positive impact on course retention and overall student grades. The control group (lecture based) pass rates were found to be 71% (N=85), whereas the experimental group (active learning) pass rates were found to be 80% (N=171).

For those students who passed the class, an independent samples t-test found that average grades significantly increased (t(197) = 2.15, p < .05), with the average grade in the control group at 3.0 (SD = .84) and the average grade in the experimental group at 3.30 (SD = .73).

A one-way repeated measures ANOVA was performed to compare the effect of the group (IV: Control vs. Experimental) on the pretest and postest Programming Concept Inventory scores (DV). A statistically significant effect of group was found (F(1, 147) = 4.46, p = .05, η² = .30. (Very large effect size based on Cohen(1988)[17]). Thus, 30% of the variance (very large effect size) was accounted for by the IV. Two paired samples t-tests were used to make post-hoc comparisons between conditions. There was no significant difference found between the pretest scores of the control and experimental groups. A significant difference was found for posttest scores (t(147) = -3.88, p < .01). As can be seen in Figure 1, posttest scores for the Experimental group (M=9.0, SD=2.20) were significantly higher than the posttest scores for the Control group (M=7.5, SD=2.14).

6. Conclusions

Taken together, these results indicate that the use of the active learning (i.e., flipped classroom) and the 3-pronged approach described above had a positive impact on course outcomes and student learning in our CS1 course. This is particularly important as it demonstrates that these types of approaches can provide real gains in student learning and retention when implemented at Hispanic-serving institutions (HSI) and those with large populations of 1st generation college students.

Our 3-pronged approach included:

- Peer-Led Learning Assistants in an Active Learning classroom
- Collaborative, active learning within the classroom
Combining Active Learning Approaches for Improving Computing Course Outcomes

• Online, just-in-time, tutorials as an integrated part of the curriculum

Using three metrics, our results indicate that this approach:

1. Improves overall pass rates for CS1 courses at a large HSI to as much as 80%
2. Improves student learning of programming concepts as measured by a standardized Java programming Concept Inventory
3. Increases the average grades for students in CS1 courses as measured by exam scores

Although our study found positive learning outcomes from our 3-pronged approach, there are some limitations to the study that indicate that more research is needed. Although drop and failure rates for CS1 courses are typically 50%, the pass rate of the control group in our experiment was 71%. This is likely explained by the fact that although the control group used was primarily lecture based, it did include some aspects of active learning in the classroom (15% vs solely lecture) along with peer volunteers to assist in class (albeit untrained volunteers). This may have been sufficient to boost the pass rates in the control group. In future studies, it would be helpful to also make comparisons with courses that are purely lecture based, without any use of active learning and volunteer peers in the classroom.

In addition, the present study only makes comparisons across two semesters. Future studies would be improved if data from additional semesters were included, providing a broader and longer-term outcomes.

Nevertheless, the improvement in student performance from the control group to experimental group provides evidence that our 3-pronged approach can improve student outcomes and learning in a CS1 course at large HSI institution. Further studies will be conducted to confirm these positive outcomes over time, and with a larger sample size.

[9]Alex Edgcomb, Frank Vahid, Roman Lysecky, Andre Knoesen, Rajeevan Amirtharajah, Mary Lou Dorf., 2015. Student Performance Improvement using Interactive Textbooks: A Three-University Cross-


[12] Lisa Kaczmarczyk and Craig Ziles have developed an introductory Computer Science Concept Inventory, which has been presented at SIGCSE Technical Symposium in 2006. A concept inventory is a tool that evaluates a student’s comprehension of a carefully selected subset of concepts, usually identified by experts in the field.


