Strategies to Incorporate Active Learning Practice in Introductory Courses

MOHSEN DORODCHI, LAUREL POWELL, NASRIN DEHBOZORGI, AND AILEEN BENEDICT

Introduction

Science, technology, engineering, and mathematics (STEM) discipline courses, and especially their introductory courses, are reported to have high failure rates. The long lecture-based format of teaching in introductory courses has shown very unsuccessful results. For example, historically, the drop"D", "F"/fail, and withdraw (DFW) rates have been reported to average 30% for introductory programming courses (Watson & Li, 2014). Students also complain about the time required and the fact that the introductory programming course demands more time than they have (Kinnunen & Malmi, 2006). Our interpretation of this issue is that students do not have enough time to digest the materials conceptually and relate their programming skills to analysis and algorithm development. When integrating active learning into such courses, significant improvements were reported over traditional lecturing. A thorough study by Freeman et al. (2014) showed that active learning leads to increases in examination performance (average increase of half a letter grade), while decreasing the failure rates. Furthermore, the study proved that while active learning achievements hold across all of the STEM disciplines and occur in all class sizes, course types, and course levels, it is particularly beneficial in small classes and in increasing performance on concept inventories.

This chapter is focused on introductory courses. The distinguishing factors that make introductory courses unique include: (a) they are offered in multiple sections with large class sizes, (b) they provide students with gateway courses into their major, and (c) freshmen experience college-level gateway courses for the first time as the foundation of their majors. Considering these features, along with the large size of such courses, special attention needs to be paid when teaching them. Beyond the content, faculty should focus on study skills, team skills, and skills for cooperative learning because they are essential to student learning.

The organization of this chapter is as follows:

- An overview of practiced active learning in introductory courses in different disciplines.
- An overview of cooperative learning versus collaborative/social learning.

- A presentation of the course model that we have been practicing for the past few years.
- The integration of our model along with innovative practices designed by us and other educators for teaching introductory courses in active learning. We present our class model that relies on student preparation and reflections on their learning.
- A discussion on how to use students' self-assessment and reflection on the teamwork and course materials to adjust and enhance the activities. In this section, we first discuss the data acquired from students' self-assessments and reflections to identify factors such as positivity, grit, and their impacts on their learning. Next, we discuss the design and adjustment of the course activities in a direction that helps students learn and perform more effectively. At the end of this section, we discussed the effectiveness of integrating self-assessment and reflection into introductory courses.
- A summary of the chapter and future directions for our findings.

Overview of Practiced Active Learning in Introductory Courses

In their work, Bonwell and Eison (1991) discussed opinions on the methods for active learning and the obstacles to its adoption. Some of these obstacles include instructor fears of being unable to teach sufficient content, the amount of preparation work required, and large class sizes. These hurdles, however, have not proven insurmountable. Active learning has demonstrated its effectiveness across disciplines and has demonstrated that it can improve both the performance and satisfaction of students.

According to Prince's (2004) survey, support can be found for multiple variations on active learning in engineering classrooms. Prince began by addressing the challenge of quantifying student classroom experiences to compare different techniques. He discussed active learning, cooperative learning, collaborative learning, and problem-based learning in engineering classrooms. After examining studies, he concluded that evidence supports the effectiveness of all these methods. Collaboration and cooperation promote positive outcomes and should be encouraged in the classroom.

Student satisfaction can be dramatically improved by incorporating active learning methods into a classroom. In Armbruster, Patel, Johnson, and Weiss (2009), students were dissatisfied with the traditional lecture-based course model in an introductory biology class. Because of this, instructors chose to reorganize the course according to active learning principles. They added problem-based activities to their lectures that students were asked to solve in small groups that would remain together for the semester. Another method to encourage students to participate more actively was the addition of "clicker questions" to each lecture that counted toward a student's participation grade. The classes were also given weekly quizzes, administered using "clickers," that were worth a small portion of the final grade. To measure the effectiveness of these methods, student evaluations and class performance were compared across three years, with the earliest year being taught in a traditional lecture style and the latter two years using active learning methods. Student performance improved, as did student satisfaction. Interestingly, while some students gave negative comments about the use of the weekly quizzes, other students listed the quizzes as highly helpful. The writers believe that this, combined with student comments, showed evidence that the quizzes made the students more aware of their own learning.

Active learning can be used effectively in large classes. Deslauriers, Schelew, and Wieman (2011) conducted a study on a first-year physics course. Eight hundred fifty students were divided among three sections in large, theater-style lecture halls. This course had been taught in a traditional lecture format using PowerPoints, instructor demonstrations, and "clicker" quizzes, which were not paired with discussion, to evaluate overall student understanding. During the 12th week of the course the authors conducted an experiment. In order to have a direct comparison of the two participating sections, one author taught the 12th week material as usual, while in the other section the material was taught by two of the authors. They integrated multiple active learning strategies, including assigning reading before class, which for consistency was also assigned to the control section, giving quizzes on that reading, and soliciting feedback in class. Additionally, they added a discussion component to the clicker quizzes and had students work in small groups to prepare written answers to questions. A test was prepared and administered to both classes when the experiment concluded. Students in the experimental section performed better on the test and, according to evaluations, were more satisfied with the material.

Instructors in a course on renal pharmacology experimented with a flipped classroom model combined with a more active teaching style (Pierce & Fox, 2012). They assigned students to view recordings of lectures before coming to class for activities. The in-class activities used a Process Oriented Guided Inquiry Learning (POGIL) design and were oriented around applying what they had learned from the videos to perform patient care. The instructors also required students to answer questions and perform calculations during the activity. Student evaluations were favorable, and student scores on the final exam improved under the new methods. The POGIL method encouraged students to consider applications of their studies, and it better prepared them for the final examination.

In contrast to the science and engineering classes discussed in this overview, McCarthy and Anderson (2000) used active learning in a history class and a political science class. Instead of a traditional lecture and discussion, the researchers used a role-playing exercise in a history class and an activity completed in pairs in a political science class. Students in more traditional classes were compared to the experimental classes, and the experimental classes performed better.

The positive impact of active learning across introductory courses is difficult to quantify, but it is equally difficult to deny. It has been used in classes across STEM disciplines and in humanities classes. While the exact methods do and must vary, the importance of focusing the student on the learning is well supported.

Cooperative Learning versus Collaborative/Social Learning

Active learning is generally defined as any pedagogical practice that engages students in the learning process. Active learning helps both student cognition and metacognition by engaging students to do meaningful learning activities and to think about what they do (Bonwell & Eison, 1991).

In research, active learning is often referred to by other names such as cooperative learning or collaborative learning. It is a challenge to provide a universally accepted term for active learning because there are multiple active learning techniques, and educators adopt different names based on their methods. However, it is possible to provide some generally accepted definitions and to highlight distinctions in how common terms are used (Prince, 2004).

In particular STEM fields, student engagement in active learning is mainly in the form of teamwork. Teamwork plays two important roles in active learning; one is peer learning that helps students learn from each other, and the other is about improving social skills. Teamwork in active learning has been introduced using different terms such as cooperative learning or collaborative learning, depending on the point of the teamwork.

Collaborative learning refers to any pedagogical technique that requires students to work together in small groups to accomplish a common goal (Prince, 2004). Therefore, collaborative learning can include all group-based instructional methods, including cooperative learning (Millis & Cottell, 1997; Smith & MacGregor, 1992).

However, some researchers believe there are distinctions between collaborative and cooperative learning and that they have different philosophical roots (Bruffee, 1995; Panitz, 1999). In either definition, the core of collaborative learning is the emphasis on student interactions rather than on learning activity.

Cooperative learning is defined as a structured teamwork where students pursue common goals while being assessed individually (Millis & Cottell, 1998). This type of teamwork has five specific aspects: individual accountability, mutual interdependence, face-to-face promotive interaction, appropriate practice of interpersonal skills, and regular self-assessment of team functioning.

While different cooperative learning models exist (Stahl, 1994; Slavin, 1983), the common element across them is a focus on cooperative incentives rather than competition to promote learning.

Based on given definitions, we believe collaborative learning can be more applicable in introductory-level courses in which students need to interact with each other and learn more social skills while learning from each other in a socially supported environment. This social aspect is important because it improves students' communication skills and prepares them for upper-level classes and the professional environment. It also makes learning a fun experience while motivating students to be actively involved due to social pressure in teams. This type of teamwork best suits less challenging and sophisticated concepts where students get a chance to learn from peers and fill the gaps between team members' backgrounds.

On the other hand, we believe cooperative learning on a larger scale can be applied in higher-level classes and capstone courses where a project is defined as a common goal and all team members apply their knowledge to accomplish the goal and develop a final product. In this form of teamwork, less emphasis is on learning and more on applying what they learned. Cooperative learning is more structured, and usually team members have assigned tasks and roles. It is more suitable for advanced and challenging topics where tasks are distributed among the team members.

An Active Learning Introductory Course Model

The teaching paradigm has been shifting from traditional lecture-based teaching methods to designing learning experiences, processes, and environments for students (Duderstadt, 2008). An integrated course design starts with analyzing the "situational factors," followed by formulating the "learning goals" as well as designing the "feedback and assessment procedures." The design and selection of the teaching/learning activities fulfills this process (Fink, 2003). Activity-based active learning seems to be one of the desirable delivery methods for such teaching and learning activities, providing both excitement and fun while emphasizing learning (Bonwell & Eison, 1991; Dorodchi & Dehbozorgi, 2017; Sanders, Boustedt, Eckerdal, McCartney, & Zander, 2017; Xinogalos, 2016). Considering the above factors, a coherent course would have a complete alignment among the activities, the assessment, and the learning goals and outcomes (Dorodchi et al., 2018). In addition, the need for an educational measurement of student knowledge aligned with activities and learning goals that goes beyond traditional tests and the methods to make inferences about student learning are instructionally essential in this model of course design (Pellegrino, Chudowsky, & Glaser, 2001).

One major issue in implementing active learning and a flipped classroom is making sure students come to class prepared. For the introductory courses, our experience showed that the preparation materials should be well structured and in line with the class activities. Therefore, in our course model, we have adapted Kolb's experiential learning cycle as shown in Figure 2.1 to make sure that the course materials and activities are developed with proper relevancy and consistency. The model visualizes Kolb's traditional four-stage model including preparation before the class, the class activity, and the postclass activities followed by the reflective observation. By replicating the reflection stage multiple times throughout the cycle, as shown in Figure 2.2, students are directed to properly conceptualize a particular course objective at different levels.

In other words, instead of having reflection only once as the last stage of Kolb's model, our model has three distinct stages with reflection integrated throughout. Students plan for the upcoming week with prep work assignments, have active learning experiences in class, and then extend their learning on assignments at home. By integrating reflection throughout all these steps, our model becomes a three-stage process with reflection: prep work that helps students prepare for upcoming classes, in-class activities to further learn and experiment on the material, and postclass assignments that extend concepts from the course so students reflect on each of these aspects multiple times throughout their learning.

Such a model also provides opportunities to design the activities in such a way to both challenge students step-by-step while encouraging them to enjoy the group and social aspects of team activities (Dorodchi et al., 2017). When designing activities for each phase and creating tests for assessing students' learning, Bloom's taxonomy was also applied. An example of one week's iteration is shown in Figure 2.3.

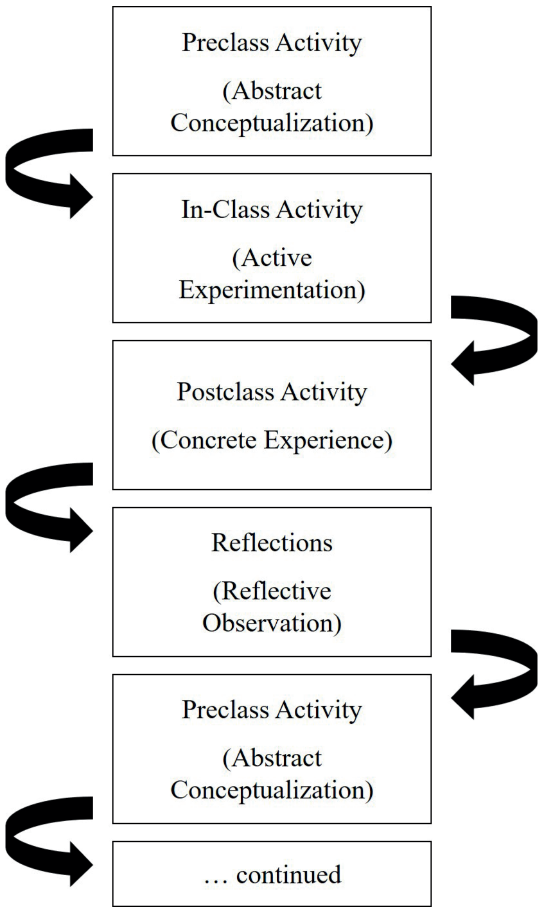

Figure 2.1. The iterative learning model based on Kolb's learning model (Kolb, 1976; Kolb, 2015).

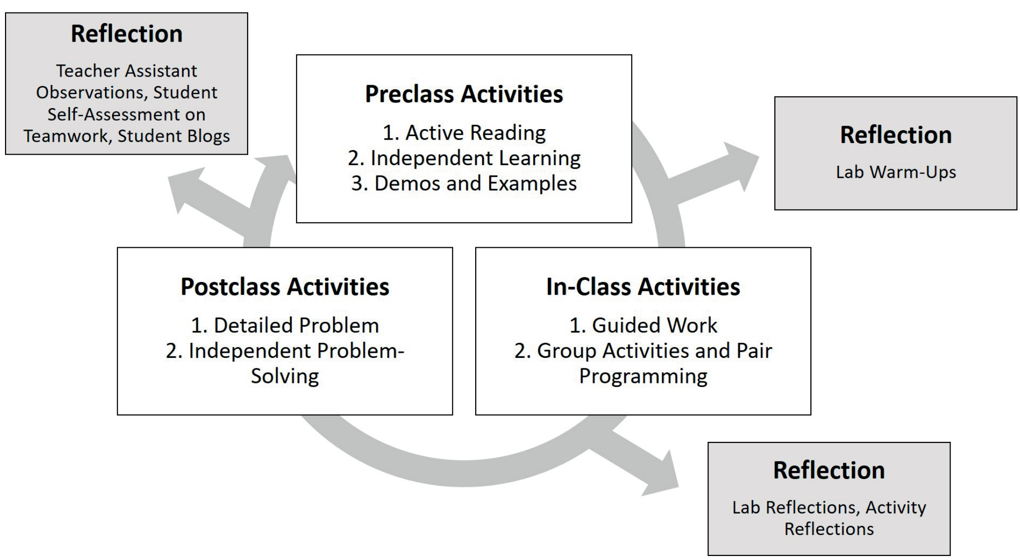

Figure 2.2. Our course model with integrated reflections.

How to Comment

You will also need to know how to properly comment for next Lab.

The icon to the left will be used a lot in Lab 2 as a reminder to comment. Follow the steps below to learn more about commenting.

...

↓

1. About how long did this work take you (pre- and postlab) *(Enter time)*

2. List two things you have learned from pair programming. Describe them briefly. *(Short Answer)*

3. How do you declare an integer as a primitive data type? *(Multiple choice)*

4. Proper commenting is just to make things look nicer, but it's not required. *(True/False)*

5. What must be imported to use the Scanner? *(Multiple choice)*

...

↓

Activity 1: Declaring Variables to help Children learn Geometry

You are asked to write a program for elementary school children to play with and learn about geometry. The program will teach them about shapes, areas and perimeters. As the start, we want you to just write a program about rectangles.

1. Start with your **block comment**!

...

↓

1. Which activities did you finish on/before the checkpoints? *(Select all)*

2. After Lab 2, how confident are you in your mastery of variables, math operations, and user input? *(Likert Scale)*

3. How was your experience with pair programming? *(Select all: 😄 Excellent, 🙂 Moderate, 😑 Bad, ...)*

4. Why is using math operations and variables important in programming? *(Short answer)*

...

↓

Program 1: Convert Pounds into Kilograms

Write a program that converts pounds into kilograms. The program prompts the user to enter a number in pounds, converts it to kilograms, and then displays the result. One pound is *0.454* kilograms.

...

Figure 2.3. Illustrating the flow of one week's lab cycle, taking sample instructions and questions from a preclass activity, reflection, in-class activity, reflection, and postclass activity.

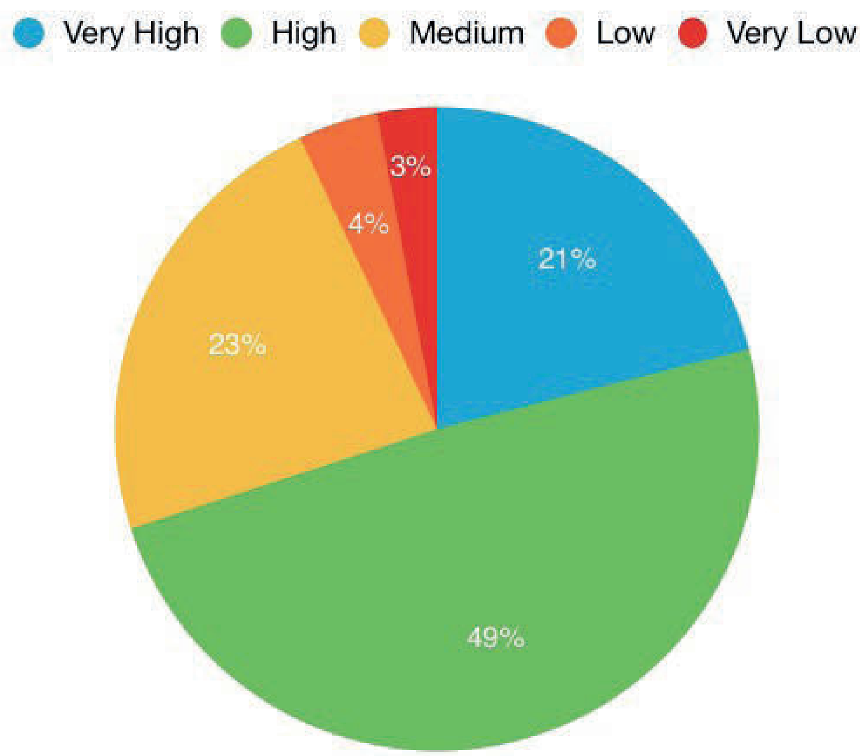

Figure 2.4. Students' perceptions about how much they learned about Java programming.

First, the students complete the prelab homework (a type of preclass activity, as shown in the cycle in Figure 2.2) meant to prepare them before coming to the lab as shown in Figure 2.3. Then, at the beginning of the lab, a "lab warm-up" is completed (a type of reflection) that asks questions relating to their preparation work. The lab itself is a type of in-class activity where students complete programming activities to strengthen their knowledge of what was learned that week. After lab, students complete the lab reflection (another type of reflection), responding to questions about the lab itself and their own learning experiences. Finally, students complete a postlab assignment (a type of postclass activity) independently for homework to further review what they have learned. Many of our own reflective prompts were aimed at having students think about their own learning experiences, hence metacognitive attributes can be included in our data. Figure 2.4 shows an example of a question from one of the reflections that asks the students near the end of the course "How much do you think you learned about Java programming?" As shown in this case, very small percentages claim they did not learn anything. Most feel they learned something or learned very well.

The reflection provides opportunities for students to talk about their learning at different levels. We think it also encourages students to think about their progress and level of understanding throughout the course. Some of our reflective prompts were designed in this way. However, we did not only include reflections just a few times throughout the semester. Instead, it was an iterative process that occurred much more frequently, such as at the end of many in-class activities. Furthermore, we did not limit our prompts to students' attitudes toward their learning experiences but asked a mixture of questions regarding many different aspects of their learning experiences.

Designing the Activities for Activity-Based Active Learning

In this model of the introductory course, we have a combination of lectures and labs. Each class or lab session includes several activities done in small groups with a duration of about 15 to 45 minutes per activity. In this section, we discuss how we designed the activities. We intended that students would do multiple activities focusing on one aspect of the course objective in a class period or lab from fundamentals to more challenging levels. We also provided opportunities to design different forms of activities to keep students interested. We call this model of activities "staged diversified activities" as explained below.

Staged Diversified Activities

As mentioned before, our course follows an activity-based active learning structure emphasizing collaborative learning that includes a spectrum of activities. Primarily, we start with warm-up activities closely related to the preparation activity before the class. We gradually transition the activities to more challenging ones using diverse methods, such as clicker-style quizzes, LMS (learning management system) individual and group work, pair programming, scratch-off quizzes, etc. We have observed that students learn better by diversifying the activities. We ask students to think and discuss the major concepts of the course while doing the activities. The goal is to facilitate the learning through transitioning from simple to more challenging concepts as well as using staged diversified activities. By scaffolding the activities, we can provide a smooth flow of concepts and make sure that students work together efficiently, implementing peer instruction and learning. Furthermore, switching between types of activities is helpful to sustain student interest. Diversifying activities prevents the effect of "getting used to it." Our observation showed that students face a plateau in their learning after a few weeks into the semester without a well-prepared, diverse set of activities. Finally, we split the activities into two major parts, leaving time in between for a Kahoot! quiz to discuss the major points of activities while assessing their learning.

In addition, we propose infusing different activity checkpoints into the activity-based active learning (ABAL) class to maintain the pace and keep all groups of students synchronized. Activity checkpoints are good times for discussions in the class. For introductory courses, activity checkpoints are essential to keeping all students engaged and on the same pace as explained here.

Activity Checkpoints

Given the time limit of each class and the need to complete many different activities per session, we use activity checkpoints. Checkpoints are spread throughout activities mostly to provide demonstrations or feedback from the instructor and to make sure all the students in large classes are learning at the same pace.

One guideline for scaffolding activities is to break tasks into smaller, more manageable pieces with feedback at the checkpoints (Silver, 2012). Both the switching of activities and the use of checkpoints allow us to enforce timely completion of the different types of activities while keeping students engaged.

Estimating realistic time limits for activities is another challenge in ABAL classes. We have evolved a scheme over time based on trial and error while testing and timing the activities ourselves, observing the students in class, and evaluating their feedback. The following brief summary of two case studies shows how we implement the discussed activity scaffolding.

Case Study 1: Checkpoint in Search-and-Sort Activity on LMS

In this class activity, students worked in pairs to complete a guided activity inside the LMS on one of the advanced and major concepts of the course. By this point, around the end of the course, the major constructs of programming, including looping, methods, and arrays, have been covered in detail with good repetitions, and students are ready to learn the search-and-sort algorithms. Many computer science algorithms require students to know the search-and-sort algorithms in depth. The activity is split into several sections, labeled "Section 1: Searching" and "Section 2: Sorting" and separated by checkpoints with proper timing per section. At the time of the checkpoint, the instructor provides overall class discussions to review some of the essential concepts with students as a short, on-demand lecture and/or a Kahoot! quiz. This gives the opportunity to adapt to the students' current needs and fill in any gaps in their learning. Breaking up the activities also helps the students deal with manageable portions of the content. In addition, the instructor has a chance to assess student learning before moving on to the next activity.

Case Study 2: Class Discussion Through Clicker-Style Quizzes (e.g., Poll Everywhere, Kahoot!)

Clicker-style short quizzes are offered between activities and sometimes during the activity checkpoints. These are used to assess student understandings of the completed class activities, provide low-stakes practice problems, and measure the effectiveness of our activity and checkpoint breakups. In hopes of reducing the performance gap between the practical and theoretical components of our course, we have designed a series of Kahoot! quizzes to provide students with constant and consistent feedback and practice. With more consistent assessment of the learners' knowledge and experiences, we are better able to determine and meet the students' learning needs.

Throughout iterations and with constant formative feedback from the quizzes, we were able to tweak the relevant in-class activity content and design questions to follow a smooth scale of easy questions building up toward more challenging ones following the algorithm (Dorodchi et al.,, 2017). Below, we showcase an example sequence of questions used in a clicker-style quiz:

- Question 1: What is scope?
- Question 2: What does the snippet below display?

```java
for (int i=0; i < 5; i++) {
    System.out.println(i);
}
System.out.println(i);
```

Figure 2.5.

- Question 3: What does the snippet below display?

```java
int i = 0;
for (i = 4; i < 5; i++) {
    System.out.print(i + " ");
}
System.out.println(i);
```

Figure 2.6.

- Question 4: What does the snippet below display?

```java
for (int i = 0; i < 5; i++); {
    System.out.print(i + " ");
}
```

Figure 2.7.

Before completing the above quiz, students would have completed the relevant activities. Question 1 is a simple and conceptual problem that reviews the students' ability to recall and understand the definition of scope. Question 2 scales up in difficulty and applies scope into a loop problem related to the most recent in-class activity completed. Students are expected to synthesize their understanding of scope and their practice in the recent in-class activity to answer this question. Students are expected to notice that the variable *i* is used out of scope. Question 3 showcases an example where the variable *i* is properly used to avoid the scope issue. By asking various questions relating to the same concept—scope—we can further reinforce the learning of students, including those who have gotten the previous question wrong. It is important to note that the instructor can see the progress of the class (including percentages of correct and incorrect responses) and thus provide additional explanations in between questions. Question 4 gets even more complex by quizzing students on for-loop exceptions with semicolons; students must first notice the misplaced semicolon and then also understand the complications caused by this misplacement. Question 4 combines the concepts from all three previous questions (for-loops and scope of the iterative variable).

In this section, we have covered our scaffolding strategy for our Kahoot! quizzes. These quizzes assist in identifying the learning gaps between our class activities and our lecture test concepts. Furthermore, it provides opportunities for more consistent assessment and feedback loops. The findings gained by witnessing the students' overall knowledge and where they may be struggling provide us the insight to tweak in-class activities and better scaffold Kahoot! quizzes. We can then build a bridge between the concepts learned in class and the concepts students are tested on in the lecture tests.

Student Reflections

Reflection is generally described as the process of giving meaning to experiences. It is a deliberate process in which people "capture their experience, think about it, and evaluate it" (Boud, Keogh, & Walker, 1985). As explained before in the learning model for the course, students need to reflect on their learning to conceptualize the major course contents.

In addition, reflection can also be used to discover the factors that make the most impact in teaching as well as hidden issues. The instructor needs to be aware of student learning experiences to support evidence-based teaching. In teaching, sometimes the instructor may go by "intuition" rather than evidence, and such intuition may not always be relevant (Guzdial, 2015). The evidence-based model in education is defined as "the integration of professional wisdom with the best available empirical evidence in making decisions about how to deliver instruction" (U.S. Department of Education, 2002). Therefore, to improve overall performance in introductory courses using active learning, we need to first discover the factors that affect student success and failure. Based on such discoveries, we may start to make improvements in our pedagogy.

We particularly investigated reflective writing as a way to have students reflect on their learnings, course content, and group work. In reflective writing, students think and write about their learning experiences to glean insight from them. Reflection is beneficial for students as it helps them to think more critically about their experiences and to challenge their own assumptions (Mezirow, 1990).

Many examples exist in engineering and computer science (CS) education literature about integrating reflective practices into courses and learning environments. Turns, Sattler, Yasuhara, Borgford-Parnell, and Atman (2014) provide a variety of examples from engineering and have created a framework for thinking about elements of reflection. Others have created pedagogies that include student reflective practices for engineering (Adams, Turns, & Atman, 2003; Shekar, 2007) and CS courses (Dorodchi et al., 2018). All these works indicate that the reflective practice not only benefits the student but also can also benefit instructors and administrators by improving at-risk classification and time to predict at-risk students (Dorodchi et al., 2018).

Statistical Analysis of Reflections

In this segment, we show descriptive statistics of the collected reflections to demonstrate results of some of the reflections as well as how the analysis can help with the scaffolding process. Analyzing reflections to gain insights into the student learning experiences can help us adapt

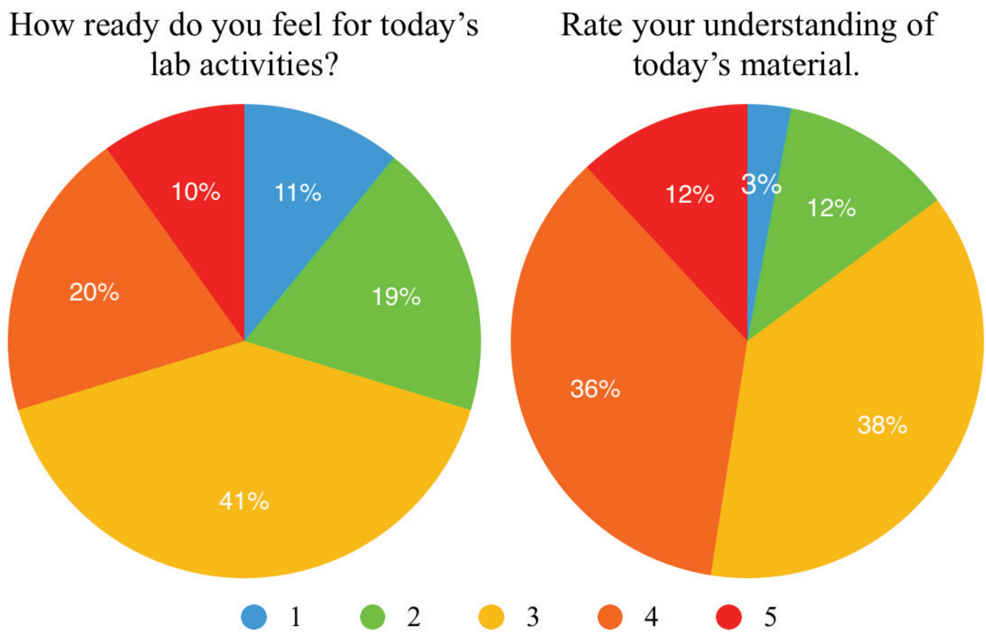

Figure 2.8. Comparison of students' perceptions before and after the lab activities.

class activities and discussions to the students' needs. Figure 2.8 is an example of a pretest/posttest practice that compares the students' perspectives about their level of readiness before the start of the lab session to their assessment of their own understanding at the end of the lab. The combined data of fall 2017 and spring 2018 semesters indicate that only 15% of students felt less competent at the end of the lab while that number at the beginning of the lab was about 30%. Such data provide useful feedback to instructors as far as the effectiveness of lab or class activities.

In Figure 2.9, students' level of interest toward programming at the beginning of the course is compared to their level of interest toward the entire computer science discipline at the end of the course. These pre/post questions were designed because it has been reported that some students quit the computer science discipline due to the difficulty of programming courses. As shown, the high level of interest changed from 40% to 54%.

The level of student confidence in performing programming tasks is also an important point of concern for faculty who teach programming courses. In Figure 2.10, using two posttests, we measure the change in the level of students' confidence from their own perspectives on two highly reported (as indicated by unigrams) challenging topics (i.e., methods and arrays) after the corresponding lab activities. This data can help evaluate the implemented scaffolding and provide information for possible additional activities to balance out the students' levels of confidence, if necessary.

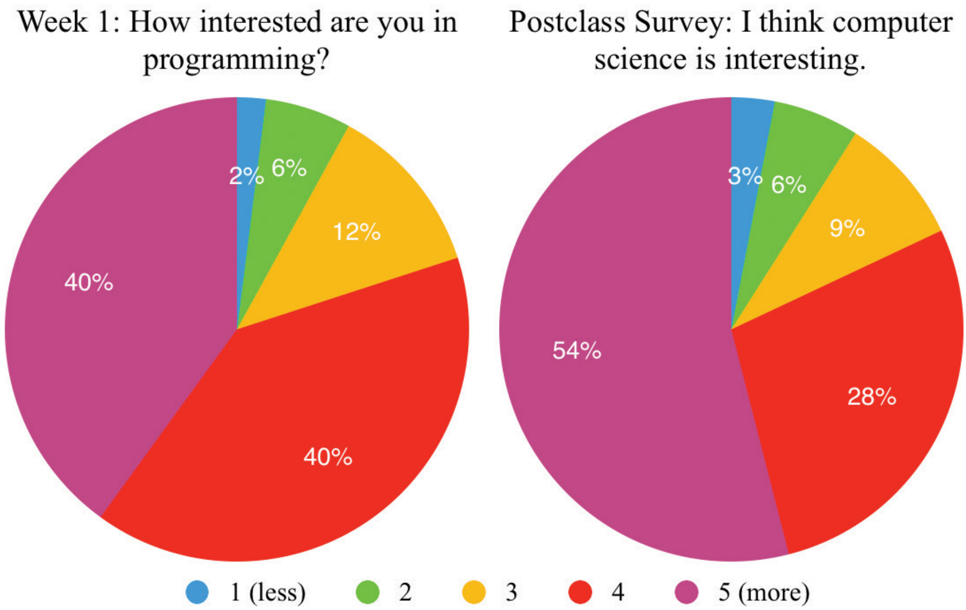

Figure 2.9. Comparison between student interest in programming and computer science at the beginning and the end of the course.

Reflection provides substantial information about students' learning; however, immediate analysis of this information is quite challenging. More elaborate analysis of the reflections is discussed in another chapter in the book called "A learning analytics approach to assessing student risk in active learning."

Reflection on Teaching Using This Active Learning Model

We have been practicing using active learning for a while and have been through different levels and a number of iterations.

Working in groups, while often advantageous, is not always easy for students. One of the advantages of this model of active learning is its flexibility in handling different student needs for group interaction. As all students are responsible for turning in their own work and can complete the majority of it on their own, students who do not want to work with their group members or who have group members who are behaving poorly are not unfairly penalized. This creates an environment that encourages collaboration without making students resent forced grouping or tying the grade of active students to their nonparticipating peers.

The feedback from the clicker-style quizzes such as Kahoot! or Poll Everywhere quizzes is another important aspect of this model. After or during an activity, students are often reluctant to ask conceptual questions. This makes judging difficult if more clarification or a minilecture

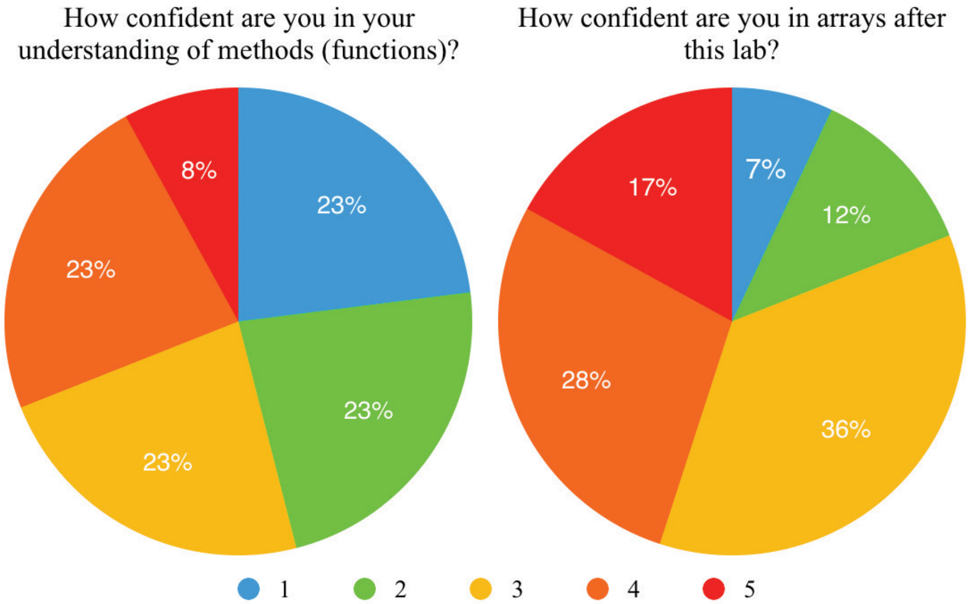

Figure 2.10. Students' perceptions about two challenging areas in computer science after the first exposure in the labs, rated from least (1) to most (5) confident.

is needed. Seeing the results makes it easy to supplement the explanation if students are struggling and helps students understand where they need help without the pressure of an exam.

Conclusions and Recommendations

Active learning in introductory courses presents unique challenges, such as unbalanced background related to the course content and team-based activities in addition to the level of motivation. A lack of homogeneity in students' familiarity with group work and the content may cause effects, such as overwhelming other students. For this purpose, the proper level of preparation for students is very critical. Such prep work has to be offered regularly with the right challenge level and amount of work. The activities and the after-class activities should be in the same direction with the prep work in more depth. Frequent repetition of the course materials and the main course objectives is necessary to remind students of critical concepts and also of the relationship of the materials with each other. The integration of continuous reflections into the course will direct students into the generation of knowledge and conceptualizing the main contents. In addition, it provides necessary feedback to the instructor to track student progress and possible struggling points in their learning, either individually or in small or large groups.

References

Adams, R. S., Turns, J., & Atman, C. (2003). Educating effective engineering designers: The role of reflective practice. *Design Studies, 24*(3), 275–294. doi:10.1016/S0142-694X(02)00056-X

Armbruster, P., Patel, M., Johnson, E., & Weiss, M. (2009). Active learning and student-centered pedagogy improve student attitudes and performance in introductory biology. *CBE Life Sciences Education, 8*(3), 203–213. doi: 10.1187/cbe.09-03-0025

Bonwell, C. C., & Eison, J. A. (1991). *Active learning: Creating excitement in the classroom* (ASHE-ERIC Higher Education Report No. 1). Washington, DC: The George Washington University, School of Education and Human Development. Retrieved from https://files.eric.ed.gov/fulltext/ED336049.pdf

Boud, D., & Falchikov, N. (1989). Quantitative studies of student self-assessment in higher education: A critical analysis of findings. *Higher Education, 18*(5), 529–549. doi: https://doi.org/10.1007/BF00138746

Boud, D., Keogh, R., & Walker, D. (1985). *Reflection: Turning experience into learning*. New York, NY: Nichols Publishing Company.

Bruffee, K. A. (1995). Sharing our toys: Cooperative learning versus collaborative learning. *Change: The Magazine of Higher Learning, 27*(1), 12–18. doi: 10.1080/00091383.1995.9937722

Deslauriers, L., Schelew, E., & Wieman, C. (2011). Improved learning in a large enrollment physics class. *Science, 332*(6031), 862–864. doi: 10.1126/science.1201783

Dorodchi, M., Benedict, A., Desai, D., Mahzoon, M. J., MacNeil, S., & Dehbozorgi, N. (2018). Design and implementation of an activity-based introductory computer science course (CS1) with periodic reflections validated by learning analytics. *2018 IEEE Frontiers In Education Conference,* 1–8. doi: 10.1109/FIE.2018.8659196

Dorodchi, M., & Dehbozorgi, N. (2017). Addressing the paradox of fun and rigor in learning programming. *ITiCSE '17 Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education,* 370. doi: 10.1145/3059009.3073004

Dorodchi, M., Dehbozorgi, N., & Frevert, T. (2017). "I wish I could rank my exam's challenge level!": An algorithm of Bloom's taxonomy in teaching CS1. *2017 IEEE Frontiers in Education Conference*. doi: 10.1109/FIE.2017.8190523

Duderstadt, J. J. (2008). *Engineering for a changing world: A roadmap to the future of engineering practice, research, and education*. Ann Arbor, MI: The Millennium Project, University of Michigan.

Fink, L. D. (2003). *A self-directed guide to designing courses for significant learning*. Retrieved from https://www.deefinkandassociates.com/GuidetoCourseDesignAug05.pdf

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences, 111*(23), 8410–8415. doi:10.1073/pnas.1319030111

Guzdial, M. (2015). Bringing evidence-based education to CS. *Communications of the ACM, 58*(6), 10–11. doi: 10.1145/2754947

Kinnunen P., & Malmi, L. (2006). Why do students drop out of CS1 courses? *ICER '06: Proceedings of the second international workshop on International Computing Education Research,* 97–108. doi: 10.1145/1151588.1151604

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *IJCAI '95 Proceedings of the 14th International Joint Conference on Artificial Intelligence,* 1137–1145. Retrieved from https://dl.acm.org/citation.cfm?id=1643047

Kolb, D. A. (1976). Management and the learning process. *California Management Review, 18*(3), 21–31. doi: https://doi.org/10.2307/41164649

Kolb, D. A. (2015). *Experiential learning: Experience as the source of learning and development*. Upper Saddle River, NJ: Pearson Education Inc.

Kolb, A., & Kolb, D. A. (2013). *The Kolb learning style inventory 2013 technical specifications, version 3.1, 3.2*. Retrieved from https://learningfromexperience.com/research-library/klsi-3_1-3_2-technical-specifications/

McCarthy, J., & Anderson, L. (2000). Active learning techniques versus traditional teaching styles: Two experiments from history and political science. *Innovative Higher Education, 24*(4), 279–294. doi: https://doi.org/10.1023/B:IHIE.0000047415.48495.05

Mezirow, J. (1990). *Fostering critical reflection in adulthood: A guide to transformative and emancipatory learning (pp. 1–20). San Francisco, CA: Jossey-Bass.*

Millis, B. J., & Cottell Jr., P. G. (1998). *Cooperative learning for higher education faculty*. Pheonix, AZ: Oryx Press.

Panitz, T. (1999). *Collaborative versus cooperative learning: A comparison of the two concepts which will help us understand the underlying nature of interactive learning*. Retrieved from https://files.eric.ed.gov/fulltext/ED448443.pdf

Pellegrino, J. W., Chudowsky, N., & Glaser, R. (Eds). (2001). *Knowing what students know: The science and design of educational assessment*. Washington, DC: National Academy Press.

Pierce, R., & Fox, J. (2012). Vodcasts and active-learning exercises in a "flipped classroom" model of a renal pharmacology module. *American Journal of Pharmaceutical Education, 76*(10), 196. doi: 10.5688/ajpe7610196

Prince, M. (2004). Does active learning work? A review of the research. *Journal of Engineering Education, 93*(3), 223–231. doi: https://doi.org/10.1002/j.2168-9830.2004.tb00809.x

Sanders, K., Boustedt, J., Eckerdal, A., McCartney, R., & Zander, C. (2017). Folk pedagogy: Nobody doesn't like active learning. *ICER '17 Proceedings of the 2017 ACM Conference on International Computing Education Research*, 145–154. https://doi.org/10.1145/3105726.3106192

Shekar, A. (2007). Active learning and reflection in product development engineering education. *European Journal of Engineering Education, 32*(2), 125–133. doi: 10.1080/03043790601118705

Silver, D. (2012). Using the "zone" help reach every learner. *Kappa Delta Pi Record, 47*, 28–31. doi: 10.1080/00228958.2011.10516721

Slavin, R. E. (1983). *Cooperative learning: Research on teaching monograph series*. New York, NY: Longman Inc.

Smith, B. L., & MacGregor, J. T. (1992). What is collaborative learning? In B. L. Smith & J. T. MacGregor (Eds.), *Collaborative learning: A sourcebook for higher education* (pp. 9–22). University Park, PA: National Center on Postsecondary Teaching, Learning, and Assessment.

Stahl, R. J. (1994). The essential elements of cooperative learning in the classroom. *ERIC Digest*. Retrieved from ERIC database (ED370881).

Turns, J., Sattler, B., Yasuhara, K., Borgford-Parnell, J., & Atman, C. J. (2014). Integrating reflection into engineering education. *Proceedings of 2014 American Society of Engineering Education Annual Conference*. Retrieved from https://www.asee.org/public/conferences/32/papers/9230/view

U.S. Department of Education. (2002). *What is EBE? Evidence-based education*. Retrieved from https://www2.ed.gov/nclb/methods/whatworks/eb/edlite-slide003.html.

Watson, C., & Li, F. W. B. (2014). Failure rates in introductory programming revisited. *ITiCSE '14 Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, 39–44. doi: http://dx.doi.org/10.1145/2591708.2591749

Xinogalos, S. (2016). Designing and deploying programming courses: Strategies, tools, difficulties and pedagogy. *Educational Information Technology, 21*, 559. doi: https://doi.org/10.1007/s10639-014-9341-9