Cogent Education

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/oaed20

Backward design: Integrating active learning into undergraduate computer science courses

Yin-Chan Liao & Marjorie Ringler

cogent education

CURRICULUM & TEACHING STUDIES | RESEARCH ARTICLE

Backward design: Integrating active learning into undergraduate computer science courses

Yin-Chan Liao[1*] and Marjorie Ringler[2]

Received: 28 September 2022
Accepted: 13 April 2023

*Corresponding author: Yin-Chan Liao, Department of Learning Sciences, College of Education & Human Development, Suite 750 30 Pryor St. SW, Atlanta, GA 30303, USA
E-mail: yliao3@gsu.edu

Reviewing editor:
Serafina Pastore, Education, Psychology, Communication, Università degli Studi di Bari Aldo Moro, Bari, Italy

Additional information is available at the end of the article

Abstract: This study was undertaken to address an issue of student retention and learning in undergraduate computer science (CS) courses. To improve students' CS learning experience, the goal of this case study was to support five CS faculty's integration of active learning into their teaching, using the Understanding by Design framework. We explored the CS faculty's considerations and experiences with active learning integration. The data included video conferencing recordings, syllabi data analysis, teacher interviews, and a focus group. The results showed that CS faculty considered three aspects in their active learning integration: (a) course structure and delivery format, (b) instructional approaches, and (c) authentic learning experiences for students. The CS faculty's primary challenge was ways to assess individual students' knowledge gained in team-based activities. We found a gradual adaptation of active learning in CS faculty's teaching practices and recognized that their development and change in pedagogies need time and personalized support.

Subjects: Computing; Teaching & Learning; Educational Research

Keywords: computer science education; pedagogies; case study; active learning

1. Introduction

Computer Science (CS) programs at universities in the United States are in high demand. The reality of some CS programs is that the courses are taught with a highly theoretical and scientific approach with limited application to real-life experiences, which may be one critical reason for students struggling with the learning content and dropping CS programs. Instructional approaches and student retention are concerns in CS and STEM-related programs in higher education (Giannakos et al., 2017). A shift from subject-centered to student-centered pedagogies may be the change needed to motivate and retain students in CS programs with the student retention issue. This study site in the Southeast of the United States served predominantly first-generation college students with low socioeconomic advantages. Most students enrolled resided in rural areas surrounding the university. This study was conducted in response to the concerning trend of high dropout rates among CS majors, with only 10% of students completing the CS degree after declaring the major. Upon analyzing the reasons behind this sharp decline in enrollment, the researchers recognized a pressing need to create a more unified and coordinated CS curriculum that would better facilitate the development of students' CS knowledge and skills. Furthermore, the study aimed to enhance the overall learning experience for CS students by implementing student-centered instruction.

In many cases, faculty in CS programs are employed for their content expertise and often are not prepared to teach by utilizing student-centered instructional strategies (Giannakos et al., 2017). Although active learning has been increasingly adopted in many CS classrooms as a student-centered approach to promoting learning engagement, a lack of time remained the most reported barrier for CS faculty who desired to change their teaching style (Eickholt et al., 2019; Eickholt, 2018). Yet, it is necessary to make pedagogical changes for enhancing students' learning experience to entice and retain students in CS programs. Intentional curriculum planning is necessary to scaffold and outline college content learning developmentally and incrementally. Hence, in this study, we shared our experience of redesigning some CS courses with student-centered active learning for a more cohesive CS curriculum and engaging learning experience to address the need for improving retention and graduation rates.

The purpose of this study was two-fold: (1) to identify CS faculty's considerations for integrating active learning in their CS courses, and (2) to explore their active learning implementation experiences. This study was initiated in the Fall of 2019, before the onset of the COVID-19 pandemic. Despite the unprecedented changes in instructional delivery, including the institution's adoption of emergency remote teaching (Hodges et al., 2020) and atypical teaching schedules, the study persisted throughout the pandemic. As a result, the CS faculty members who participated in this study were required to modify their course schedule from a 15-week to an 8-week format and transition from face-to-face to virtual course delivery during the study period in Fall 2020. The study's primary focus was on implementing the principles of Understanding by Design and examining CS faculty's considerations included in integrating active learning, regardless of whether the courses were offered in person or online. The following research questions guided this study:

(1) What were the CS faculty's considerations when integrating active learning in CS course redesign?

(2) What were the CS faculty's experiences with active learning integration in their courses?

2. Literature review

Thota (2014) claimed that "the long-term goals for the introductory programming course are that the students should eventually develop the problem-solving ability and design competence" (p. 130). However, many CS programs in higher education have experienced high dropout and failure rates (Bennedsen & Caspersen, 2007; Watson & Li, 2014). This challenge might stem from barriers to supporting students' CS competency development. Medeiros et al. (2019) conducted a systematic review of 89 articles published between 2010 and 2016 on challenges in teaching and learning introductory programming in higher education. They highlighted that the most stated key to learning programming, especially for novice learners, is problem-solving skills, including "understanding the context of a problem, identifying key information, and making a plan to solve it" (Medeiros et al., 2019, p. 80). However, students often struggled with identifying problems, expressing solutions, and debugging. To improve student engagement and learning outcomes in higher education, Chasteen et al. (2011) proposed course redesign as a way of transforming and restructuring a course by changing instructional approaches and learning resources.

2.1. Active learning in undergraduate CS courses

Active learning, which was first proposed by (Bonwell & Eison, 1991) centering student engagement in learning activities, has been used as an instructional approach in various higher education disciplines, including computer science. As defined by Freeman et al. (2014), "active learning engages students in the process of learning through activities and/or discussion in class, as opposed to passively listening to an expert. It emphasizes higher-order thinking and often involves group work" (p. 8413). Typically, active learning yields an authentic experience that involves hands-on, relevant, and contextualized learning opportunities with group work inside and outside the classroom (Kovarik et al., 2022). While there has been increasing recognition of the benefits of an active learning environment in supporting CS students' motivation, engagement, and learning performances (Berssanette & de Francisco, 2021; Gao & Hargis, 2010; Vihavainen et al., 2014), how active learning has been utilized in classrooms is not yet widely studied in academia (Freeman et al., 2014). There are many ways to integrate active learning into college STEM education.

Active learning may also be the predominant method of instruction for instructors that choose to engage students in problem-based learning, project-based learning, and those that rely on group interactions in classrooms. Active learning activities allow instructors to provide real-time remediation opportunities and formative assessments during the learning process (Gao & Hargis, 2010). One study investigated an active learning college lesson about introductory STEM biology class that utilized the 5E format (Idsardi et al., 2019) for college students. This 5E format guides the instructor to plan for engagement, exploration, explanation, elaboration, and evaluation. This active learning study indicated that students were learning science and planned to be teachers who learned biology concepts in greater depth because of the active learning pedagogies.

Caceffo et al. (2018) studied the relationship between instructional approaches and students' learning and motivation by implementing and comparing traditional instruction (i.e., lecture-based learning) and active learning approaches (i.e., problem-based learning and peer instruction) in an introductory CS course. In the study, peer instruction was implemented as part of a flipped classroom. Students studied learning content pre-class, and the teacher identified and helped with students' misconceptions in class. Regarding problem-based learning, students discussed solutions to problems in their internal groups, followed by collaborating with other groups to finalize their problem solutions. The study results showed that students who received problem-based learning performed 16% better in learning evaluation and were 30% higher in motivation than lectured-based instruction. The students reported they learned better through peer interaction and collaboration, whereas the lecture-based class was tiring and difficult to understand. However, the authors also highlighted a challenge: the CS faculty needed to spend considerably more time preparing a class with an active learning approach than a traditional lecture-based class.

Although there has been evidence of the effectiveness of active learning in teaching (Freeman et al., 2014; Isaias et al., 2021; Vihavainen et al., 2014), barriers such as student resistance, preparation time, and teachers' efficacy of instructional techniques remain in changes in teaching practices. Finelli et al. (2018) surveyed 1,051 students in 18 introductory engineering courses where active learning was implemented to examine their perceptions of and responses to active learning. The students reported that their teachers used more explanation strategies (e.g., clearly explaining what students were expected to do for the activities) than facilitation strategies (e.g., walking around the room to assist students and provide feedback to students). The authors found a strong correlation between students' perceptions of active learning teaching strategies and their responses to active learning. Students actively participated with little resistance, had less distraction during learning activities, and evaluated the course and instructor higher when implementing active learning.

2.2. Backward design of courses

The process of instructional change in higher education is complex and needs time with appropriate support (Smith, 2012). Lectures that consist of faculty sharing information and students taking notes are not conducive to elevated levels of learning, nor are they considered highly motivating to students to learn or persevere in a program or degree. Many graduates who experience large classroom sizes that are taught by lecture often choose to leave CS programs. Engaging undergraduate students in learning programming and computing has been challenging in many other CS classrooms (Cheah, 2020; Gomes & Mendes, 2014; Hertz & Jump, 2013). Students often identify instruction as a challenge they face that results in attrition and low enrollment.

To improve the purpose and effectiveness of instructional activities and improve the student learning experience, backward design, also referred to as understanding by design (Bowen, 2017), was proposed to guide the course restructuring process, focusing on the intentionality of instruction and student learning. As defined by Wiggins and McTighe (2011), backward design is a curriculum development approach that emphasizes facilitating student learning and understanding rather than the mere coverage of learning content. Unlike traditional approaches that start with the topics or content to be covered, backward design starts with the end goal and the desired learning outcomes and works backward to identify the specific instructional strategies and content needed to achieve those goals.

Many college faculty do not necessarily utilize backward design to plan learning experiences for their adult learners simply because they typically use teacher-centered approaches to teaching such as lectures that are not effective for in-depth learning (Sadler et al., 2017). Typical instruction is assessed by midterm and final exams and is a test of memorization skills. Faculty are often concerned with cheating when their focus should be on engaging students in higher levels of thinking, problem-solving, and applying theory to practice. Fink (2003, 2013) authored a book incorporating backward design for higher education to help faculty design instruction for significant learning experiences. These significant learning experiences are designed to engage students in their learning and produce learning outcomes that last after graduation and prepare them for the world of work.

Backward design for college course design begins with developing student learning outcomes that describe significant learning. In this study, the researchers were faculty in education that worked with computer scientists to redesign their courses to include opportunities for active learning. We noted that CS faculty viewed learning as incremental and spent substantial time lecturing on basic concepts and foundational knowledge. STEM faculty often focus their teacher-centered efforts on covering content (Sadler et al., 2017). When college students apply foundational knowledge to address complex projects, they develop their ability to think critically and perform a job in this field (Fink, 2013). Backward design guides faculty to design outcomes for higher levels of thinking. Once the course outcomes are designed, faculty design evidence or assessments that show that the students achieved these learning outcomes. The learning outcomes and assessments help faculty design the teaching and learning activities to support the goals and products planned for student learning. Integrating learning goals, assessments, and teaching and learning activities is key.

3. Methods

This qualitative case study (Yin, 2014) was undertaken in the Department of Computer Science at a public university in the Southeast United States. Two education researchers led this study at their university in collaboration with five CS faculty members to adjust instructional pedagogies in CS courses. The study focused on exploring the considerations and experiences of the CS faculty members in a course redesign initiative. This initiative aimed to increase learning content cohesiveness and enhance students' learning experiences in the CS program to retain students by integrating active learning into CS courses through the backward design approach. The two researchers participated as consultants to provide faculty pedagogical suggestions during their course redesign. The researchers collaboratively utilized narratives of the recorded consulting session to identify each faculty's implementation stories in the thematic analysis (Braun & Clarke, 2006) of each of their considerations and implementation. The researchers together also reviewed the revised syllabi utilizing the Understanding by Design framework to examine the CS instructors' considerations while revising curricula and identified professional challenges that were barriers to course design and student-centered teaching practices.

3.1. Study context

At the time of the study, the average enrollment in the only undergraduate BS in CS program was 430 students. The CS program had a history of alarmingly low retention and graduation rates. The average 4-year graduation rate for the 2000–2012 cohorts was 4.5%, which increased to 10% at the 6-year graduation rate. CS became its own department in 2002, separating from mathematics. Therefore, the department's culture and approaches to teaching and learning were still influenced by mathematics pedagogies that focused on a highly theoretical approach to teaching and learning. Even though the CS department moved to the College of Engineering and Technology, the faculty operated in a silo. The College of Engineering and Technology programs experienced growth in faculty and investments, yet the CS faculty and the CS program curriculum had not changed with the monumental and rapid changes in the computing discipline. At the time of the study, the curriculum's scope and sequence had not been revised since the early 1980s, and therefore students were learning theoretical approaches and missing more authentic project experience and skill development in areas such as communication, collaboration, and teamwork. The outdated curricula have a negative impact on student success (Henderson et al., 2011). This situation was not unique to this study context and applies to CS academia (Radermacher et al., 2014).

While several factors could affect the high attrition rate in the local context, one of the major issues was that students struggled to transition from introductory to advanced courses in the CS program. The CS program collected survey and interview data from the students and faculty that indicated this issue was due, in part, to a lack of coherence in the prerequisite course content taught in the first year of the CS program. The CS department chair and faculty involved in this project presented active learning as a viable solution for faculty to implement.

3.2. Participants

Ten faculty comprised the CS department. Five (50%) faculty that participated were part of an IRB-approved NSF project and volunteered to work with education faculty on the process of curriculum mapping and Backward Design. Table 1 summarizes information about the five participating CS faculty members (in pseudonyms) and their chosen courses for the redesign. Introductory courses were required to be taken in sequence. In this study, the faculty of these three introductory courses wanted to align their course outcomes, scope, and sequence. The faculty members of the intermediate and advanced courses volunteered to participate in the course redesign process.

3.3. Data collection and procedure

The plan for the study was developed in 2019. The study was carried out for four weeks with each participating CS faculty member during the Summer of 2020. The researchers adapted the Backward Design Template (Bowen, 2017) by modifying instructional descriptions and guidelines to be relevant to the CS education context in this study (see Appendix A). Due to the COVID-19 pandemic, weekly meetings were held virtually and recorded with consent by the participants. The study timeline and activities are listed below.

Table 1. Summary of participant demographics

Name	Gender	Role	Course Redesigned	Course Level	Requirement
Jack	Male	Teaching Instructor	Algorithmic Problem Solving	Introductory	Required
Ben	Male	Teaching Instructor	Algorithms and Data Structures	Introductory	Required
Donna	Female	Tenured Professor	Data Abstraction and Object-Oriented Data Structures	Introductory	Required
Nathan	Male	Tenure-Track Professor	Organization of Programming Language	Intermediate	Elective
Vincent	Male	Tenured Professor	Digital Image Processing	Advanced	Elective

- Week 1: The CS faculty established the course student learning outcomes and identified the essential knowledge and skills students should have by the end of the course. The faculty, guided by the education faculty, pinpointed major CS concepts to articulate what the student would know and do by the end of the course.
- Week 2: The CS faculty identified student products and created assessment activities based on desired student learning outcomes, such as peer review and group presentation. The CS faculty discussed ways to integrate active learning approaches to address instructional and learning needs with the education faculty.
- Week 3: The CS faculty updated the course syllabus, including detailed information about each assessment activity. The education faculty discussed an outline of a syllabus with the CS faculty that prompted alignment among outcomes, assessments, and descriptions of major projects and activities. They also facilitated the discussion to decide on key topics to reach the course learning outcomes.
- Week 4: Same as the task in Week 3, the CS faculty continued updating their course syllabus. The education and CS faculty explored and developed active learning activities per topic. In discussions with the education faculty, the CS faculty adjusted the instructional and assessment activities to ensure alignment with the instructional goals.

Primary data sources collected included meeting recordings, course redesign documents, interviews with individual CS faculty members, and a focus group. These multiple data sources generated an in-depth description and a holistic view of CS faculty members' perspectives and reported teaching practices of redesigned courses with active learning activities (Yin, 2014). Table 2 shows the many sources of data. Course redesign virtual meetings with each CS faculty member ($n = 22$) were in the Summer of 2020. During the 2020–2021 academic year, eight redesigned course syllabi ($n = 8$) were analyzed. In addition, the researchers conducted semi-structured, hour-long individual interviews ($n = 6$) towards the end of the Spring 2021 semester to learn about the CS faculty's active learning implementations. A 45-minute virtual focus group was conducted in the Summer of 2021 to let the CS faculty members reflect on their course implementations for improvements. All the virtual meetings, interviews, and the focus group were on Microsoft Teams, audio-recorded, and transcribed.

3.4. Data analysis

We utilized Braun and Clarke's (2006) theoretical thematic analysis to focus on faculty considerations and experiences experimenting with active learning in their lessons. Braun and Clarke (2006) defined thematic analysis as "a method for systematically identifying, organizing, and offering insight into patterns of meaning (themes) across a data set," which allows the researcher to "make sense of collective or shared meanings and experiences" (p. 57). Identifying themes helped us understand the CS faculty's considerations in course redesign and experiences of active learning integration in CS courses at a deeper level.

The researchers discussed themes related to active learning and generated common themes. Without a pre-existing coding scheme, two researchers familiarized themselves with the data as the first step.

Table 2. Summary of data collected over one year

Time	Research Activities	Data Collected
Summer 2020	• Course redesign • Syllabus revisions weekly	• Meeting recordings
Fall 2020 – Spring 2021	• Active learning implementations	• Redesigned course syllabi • Individual faculty interviews
Summer 2021	• CS faculty reflection on modifications	• Focus group

Next, each researcher individually reviewed the same interview transcripts and video recordings of meetings to generate the prevalent themes centered around the research questions in a spreadsheet, followed by a discussion to compare and calibrate the themes again. After, each researcher reviewed and utilized the calibrated themes to analyze half of the data individually. Finally, the two researchers came together to discuss their thoughts for the final analysis and finalize the themes.

Having multiple data sources in addition to the interviews, such as the focus group and redesigned course syllabi, collected throughout the study allowed us to triangulate our findings to increase the credibility and trustworthiness of our findings in the qualitative data (Patton, 2014).

4. Results

Study findings indicated that when the CS faculty planned to integrate active learning activities into their courses, they expressed considerations centered around the course structure, instructional approaches, and the student's learning experience. The COVID-19 pandemic affected these considerations directly, yet the CS faculty members in this study persisted in their active learning implementations during these challenging times. In addition to these extenuating circumstances, CS faculty faced common concerns related to time to plan, time to grade, and resistance to change. These considerations are described next.

4.1. Consideration of course structure

Before this study, CS faculty focused entirely on their course content and did not consider the curriculum placement of their course within the entire program of study. After completing this study, the participating CS faculty members of the introductory courses acknowledged their responsibility in preparing students for the next level courses. Ben shared, "There is the expectation that I'm supposed to get students to a certain level, so they'll be ready for the next class. So, there's that fine line between having to review stuff that they probably didn't quite get very well in the previous class versus having enough time to cover what I need to be covering." Recognizing common struggles novice CS learners might have, Jack emphasized the foundational knowledge in his class, which was the first introductory course students take in the CS program. Jack elaborated in the interview, "Students absolutely have the hardest time understanding how to decompose problems. I have them spend a good deal of time with Jeanette Wings and videos on computational thinking. This sort of amorphous idea that they think is just words."

The COVID-19 pandemic in the Fall of 2020 forced the CS faculty to shift rapidly to emergency remote learning with a shortened teaching schedule when the study was conducted. The university mandated that CS faculty adjust their teaching timeline from 15 weeks to 8 weeks. A major consideration for this change focused on how to cover all the learning content and ensure that students meet course learning objectives. They struggled with the notion of assessing content knowledge and application together as opposed to assessing memorized concepts first and then applications. After the study, the CS faculty members recognized the benefits of shifting from traditional assignments such as midterm exams to projects to demonstrate the application of learning. In the end-of-year interview, Nathan asserted the importance of connecting assignments to objectives instead of busy work to monitor time on task.

The shortened length of the course from 15 weeks to 8 weeks seemed to limit the CS faculty's active learning integration regardless of the course level. The CS faculty expressed that group interactions and hands-on activities required more time and effort to plan and manage. Jack stated, "One size doesn't fit all with active learning. You do have to think about the amount of time you have in a class period and organize it well." Donna also reflected in the focus group that limited instructional time could hold back the integration of active learning in an 8-week teaching schedule. She explained, "With the block schedule last fall, we only had seven weeks to teach everything. Considering the exam and so on, it was really fast-paced."

4.2. Consideration of instructional approaches

When the CS faculty members redesigned their courses, they considered *how* and *when* the learning content should be taught. Their value of active learning as a student-centered instructional approach was reflected in the interviews. For instance, when Jack reflected on the redesigned activities in his course, he shared, "All CS instructors should be ready for integrating active learning such as hands-on activities and trivial exercises in small groups because students can't learn programming without programming" (focus group). Donna also stated, "I really want to do more active learning in my course because I think it can help students learn better" (interview).

Regardless, how the CS faculty members' value of active learning translated to changes in their instructional approaches in the course redesign varied. Four of the five faculty members considered and employed new instructional approaches emphasizing group collaboration (e.g., project-based learning, team-based activities, and replacing the final exam with a group project) for active learning integration. For example, Vincent integrated problem-solving projects to let students work in teams throughout the semester. His students were skeptical about the purpose and usefulness of the content they needed to learn at first, but the learning through the process was rewarding and valuable. He shared:

> Students may not have all the bits and pieces to see the big picture but will come together at later points. Here's a small-scale problem, the context, and the real application. Then we need to learn various knowledge and skills to solve this problem. So, students were working in teams and were quite excited. (Vincent, interview)

In Nathan's redesign, he had small chunks of time for lectures and activities. He shared in the focus group, "I tried to intersperse students about 10 to 15 minutes of lecture, followed by a few minutes of in-class activities like think-pair-share."

Unlike other participating CS faculty members, Donna took tiny steps toward active learning integration. In Donna's course, she emphasized individual student development through her scaffolding in instructional materials and during lab time rather than peer interactions and collaborations. Especially when the courses were forced to turn into a virtual format during the pandemic, Donna expressed in the focus group that "group activities might not be as effective in the online environment compared to face-to-face facilitation." Therefore, she redesigned her course using a "learning by doing" approach for her students to learn through small problem-solving practices to establish foundational knowledge for more complex assignments.

4.3. Consideration of authentic learning experiences for students

When redesigning CS courses for active learning integration, the CS faculty members also took students' learning experiences and outcomes into account. Specifically, they would try to ensure that the active learning activities were aligned with students' learning interests and expectations and kept them engaged in class. For instance, Jack used game development to address students' learning interests. He explained, "most students want to learn to program so they can write games. So, we have to take this fact and use it as a tool to help them learn."

Moreover, having students create a program or work on a project is another way to make student learning more authentic and applicable. Donna shared the importance of providing hands-on learning experiences for students. She said,

> If students only watch you do it [problem-solving] or listen to you, they don't learn enough. You have to have them work on it, and then they get the idea. That's a type of problem-solving activity. But it doesn't have to be big, because the small ones are especially helpful for the weaker students. (Donna, focus group)

Aligned with the instructional approach, Vincent considered group work as a means of providing students with an interactive and authentic learning experience. In particular, he stressed the importance of students collaborating with peers to solve problems, as this would prepare them for real-world CS work scenarios and enhance their readiness for future employment opportunities. Vincent integrated project-based learning and created multiple complex problems for his students to solve in teams, despite acknowledging that the time constraint in a packed course schedule was a challenge. He explained, "the reality is, the problem is so complex that students don't have enough time to go through many PBL projects in a semester. But it truly reflects a real-world scenario when you enter employment."

4.4. CS faculty's active learning integration experiences

Based on the CS faculty members' reported teaching practices and a review of their syllabi, we identified active learning activities and strategies that enhanced students' learning experiences. At the same time, we found that limited time for planning, instruction, and student assessment was a major challenge to the faculty's active learning integration in CS classrooms.

4.4.1. Adaptation of active learning

All five CS faculty members successfully developed an active learning-integrated course syllabus using the backward design approach with guidance from the researchers. Through the active learning integration project, the CS faculty began communicating with each other regarding what content and activities to be covered in their courses to provide students with a more cohesive and engaging learning journey in the CS program.

Even though the active learning implementations might not be as smooth as planned due to the pandemic, the CS faculty embraced the challenges. For instance, during the pandemic, Nathan found a way to pair up online and on-site students to work on programming exercises. He shared, "For online students, I had put them in breakout rooms. For the students that did participate in class, I asked them to pair up with the neighboring students. I would randomly pick students to present their solutions to the problems or any questions to ask each other."

Despite the challenges, the CS faculty members reflected on their teaching practices for further improvement. We discovered that they started integrating one or two new active learning activities in their teaching practices and gradually adapted the integration based on students' feedback and performance. For instance, in Ben's class, he replaced all individual work with team-based assignments that allow students to solve problems as a team with peer interaction and support. He also added peer evaluation for teamwork. While Ben seemed nervous about the new implementation at first, he reported positive outcomes at the end of the semester. Ben shared,

> Having the students submit the peer evaluation about the other group members was a great idea; having them additionally rate themselves has proven to be fantastic! They are using a great reflection tool to think about how they can contribute better to the next assignment. (Ben, interview)

After two consecutive semesters of active learning integration, Ben reflected: "Compared to the fall semester, things are going much smoother this time. Now, I have a much better idea of how these new ideas fit into the class and how they should be implemented."

The active learning integration in Nathan's course redesign was a reflective and iterative process. He modified the active learning activities in the second semester based on students' feedback in the first semester to address students' learning needs. Nathan shared his adaptation of active learning in the interview:

> In some activities, I would present a theoretical concept and an example, then ask for a problem from my students. They would ask, "Is it possible to walk through a problem beforehand so we can look and compare what we're supposed to do?" So, next time I offer

> the course, I'll have to try to spend a little bit more time on the details and provide more scaffolding for the students so they can relate.

4.4.2. Challenges of active learning integration

During the study, the CS faculty were forced to shift from a 15-week face-to-face teaching schedule to an 8-week online teaching schedule in a pandemic circumstance. This rapid shift in the course schedule and delivery format caused an unprecedented challenge for the CS faculty members to plan and implement active learning activities. Ben shared: "On the downside of it, the students that I have this semester are the ones that had the eight-week introductory course in the fall. So, they are nowhere near as prepared for my class as they should be." Jack also explained, "it's a slippery slope to try to introduce a lot of new pedagogical sorts of things . . . As you really do have to change things, 55-minute classes aren't enough to do active learning exercises in a virtual environment." On the other hand, "going online was challenging because students were not familiar with the online learning environment." The emergent remote teaching created additional difficulty for Jack to facilitate student group work: "I had students spend 30 minutes in breakout rooms for their group work in a 55-minute class, but lots of technical issues came up, such as failure of WebEx [online conferencing tool], connection issues, and unavailability of recording breakout sessions."

Another challenge to active learning integration the CS faculty members faced centered around student assessment. Some faculty found that fairly assessing individual students' knowledge and performances in teamwork was challenging. Vincent reported," Each student works on this problem by herself. So, then it forces the students to learn every aspect of the problem. You know that person is not doing the job. You might as well do it yourself if you want to know everything." With some frustration, Ben reported: "We've got some people that are probably just barely slotting by because their group ratings are not low enough to make them fail. Yet they might've contributed some, but they don't know how to sit down and write a whole program by themselves." Like Vincent and Ben, Donna expressed the challenging part of her active learning interaction: "How do I balance individual and group work? I find that challenging. I was worried that if I do group work, not every student will put effort into that."

5. Discussion and conclusions

One major takeaway from this study was the importance of creating a space for faculty to talk about teaching and learning. In this study, the backward design process successfully encouraged faculty to engage in dialogue about teaching and learning. These discussions about pedagogies and their implementation opened a world of collaboration and collegiality among faculty. In this case, we found that professors of education, the researchers, were effective in facilitating these discussions. One reason for this is that the researchers had no vested interest in directing or affecting the content of the CS program. The CS faculty members oversaw their content and would entertain researchers' questions and inquiries, yet applied aspects of the backward design and active learning activities that were most aligned with their teaching styles and their teaching agendas. It is important to note that these levels of instructional dialogue were achieved over time when an intentional process was articulated and implemented.

Additionally, the impact of this study also includes the recognition of the potential positive effect on the tenure and promotion process for CS faculty members. As teaching is highly valued in this institution, the CS faculty members were motivated to invest their time and effort in enhancing their instructional strategies. Furthermore, we suggested that the department chair can incorporate the importance of instructional innovation and improvement by reflecting on this study during the faculty's annual evaluations. It is important to acknowledge that each faculty member may interpret and apply new pedagogies in their unique ways, and promoting long-term change requires an understanding of this diversity. In this way, we can move towards significant improvements in the curriculum and enhance the quality of education provided to students.

The findings of this study suggest that minor modifications can be implemented relatively easily, while significant changes require a more extensive and prolonged effort. Furthermore, faculty members tend to adjust to changes incrementally, based on their particular circumstances. Although faculty development is a challenging task, it remains essential for enhancing teaching and learning. As noted by Borda et al. (2020), resistance to change is an inherent factor that needs to be addressed in any instructional change initiative.

The study results revealed that the CS faculty members tended to identify barriers to frequent and consistent active learning implementations. Despite this, it is essential to embrace change to ensure that the degree curricula remain relevant and responsive to the rapidly evolving STEM fields. Another takeaway from this study is the importance of establishing mechanisms for ongoing faculty reflection on student learning experience and adaptation to instructional approaches that support student engagement. Research has shown that teachers are more likely to boost their teaching efficacy and alter their teaching practices when they observe successful implementations of new instructional approaches leading to improved student learning experiences and outcomes (Opfer et al., 2011; Thornton et al., 2020). In this study, we found that curriculum design and redesign can serve as a constructive and systematic approach to continuous faculty development in teaching and learning. The relationship between faculty changes in teaching practices and students' learning is reciprocal and cyclical. Through their teaching practices and experiences with students, CS faculty can establish and shape their teaching values and beliefs (T. Guskey, 2020).

The researchers found that CS faculty members lacked exposure and training about instructional pedagogies which is often why many college faculty continue lecturing and do not try student-centered pedagogies such as active learning. In this study, researchers had expertise in active learning pedagogies, and they guided the course redesign process. The Backward Design process, well recognized in the education profession, was adapted to CS education, and co-constructed through ongoing coaching and professional development. We found that faculty development was vital in addressing the issues of student retention and learning achievements in CS education (Beach et al., 2016). We find that intentional improvements in instructional pedagogies are an effective vehicle for faculty development. This faculty development will contribute to improving student learning and engagement (Condon et al., 2016; Kinzie et al., 2019). We experienced interdepartmental collaboration around teaching and learning and we recommend that college faculty study the considerations in this article when addressing change in pedagogies.

The final takeaway from this study highlights the challenges associated with shifting mindsets toward assessing student learning. The traditional approach of assessing learning through multiple-choice tests or quizzes was deeply ingrained, making it difficult for some CS faculty members (e.g., Ben and Donna) to embrace new approaches. Concerns were raised about the potential for cheating, particularly with group work or project-based assessments, because they felt that memorization was necessary before practical application. Despite these challenges, fruitful discussions ensued, and the CS faculty were encouraged to try active learning activities. While not all CS faculty utilized active learning as a measure of learning, they did find that active learning was an effective way to gauge student engagement and learning in applied contexts. Faculty members, such as Vincent and Nathan, also shared their experiences with colleagues, which piqued interest and led to more experimentation with active learning. Ultimately, the researcher observed that CS faculty who continued to utilize backward design and embraced active learning as a measure of learning were able to see the broad benefits of this approach, and their experiences may help pave the way for future changes in assessing student learning.

Finally, the COVID-19 pandemic has brought about significant changes in CS education, as highlighted in an article by Fraser and Mancl (2021). It is recommended that educators adapt to these changes by embracing online teaching methods and leveraging technology to create engaging and interactive learning experiences for students. Furthermore, it is crucial to ensure that online learning is accessible and inclusive for all students, including those from diverse backgrounds. In the post-pandemic era, it is essential for CS faculty to be adaptable to online and in-person instruction and facilitate a sustainable learning environment to maintain student engagement in CS education (Gill et al., 2023). This can be achieved through the use of an action research methodology that involves continuous reflection, evaluation, and improvement of teaching and learning practices. Additionally, it is recommended to incorporate sustainability principles and diverse perspectives, and collaborative learning opportunities into the curriculum to prepare students for the evolving job market and ensure that they are equipped with the knowledge and skills needed to address global challenges (Gill et al., 2023; Mozelius, 2022).

6. Limitations

The unprecedented COVID-19 pandemic created a unique study context for both the researchers and participants, as they had to switch to a new instructional format. This context may have influenced the results of this study that examined the challenges faced by the CS faculty members in implementing active learning. However, the CS faculty members' course redesign using the backward design approach was well-documented, thanks to the ability to record sessions and revisit conversations and activities. Despite the pandemic, this study revealed that the CS faculty chose to opt into learning and implementing active learning. Therefore, projects like this should start small with faculty who are willing to experiment with student-centered pedagogies. It is also crucial to provide ongoing, personalized coaching support and build trust among faculty, especially if they are from different program areas. By taking these considerations, it is possible to shift from content-centered to student-centered pedagogies and benefit the students in CS education.

Funding
This work was supported by the National Science Foundation under Grant # 1730568.

Author details
Yin-Chan Liao[1]
E-mail: yliao3@gsu.edu
ORCID ID: http://orcid.org/0000-0001-8854-7809
Marjorie Ringler[2]
ORCID ID: http://orcid.org/0000-0003-2484-3873
[1] Department of Learning Sciences, Georgia State University, Atlanta, Georgia, USA.
[2] Department of Educational Leadership, East Carolina University, Greenville, USA.

Disclosure statement
No potential conflict of interest was reported by the authors.

Ethics Approval
This study was reviewed by the Human Subjects and Institutional Review Boards of East Carolina University and received expedited approval (UMCIRB 17-000947).

Citation information
Cite this article as: Backward design: Integrating active learning into undergraduate computer science courses, Yin-Chan Liao & Marjorie Ringler, *Cogent Education* (2023), 10: 2204055.

References

Beach, A. L., Sorcinelli, M. D., Austin, A. E., & Rivard, J. K. (2016). *Faculty development in the age of evidence: Current practices, future imperatives*. Stylus Publishing, LLC.

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin*, *39*(2), 32. https://doi.org/10.1145/1272848.1272879

Berssanette, J. H., & de Francisco, A. C. (2021). Active learning in the context of the teaching/learning of computer programming: A systematic review. *Journal of Information Technology Education Research*, *20*, 201. https://doi.org/10.28945/4767

Bonwell, C. C., & Eison, J. A. (1991). *Active learning: Creating excitement in the classroom*. Jossey-Bass.

Borda, E., Schumacher, E., Hanley, D., Geary, E., Warren, S., Ipsen, C., & Stredicke, L. (2020). Initial implementation of active learning strategies in large, lecture STEM courses: Lessons learned from a multi-institutional, interdisciplinary STEM faculty development program. *International Journal of STEM Education*, *7*(1), 1–18. https://doi.org/10.1186/s40594-020-0203-2

Bowen, R. S. (2017). *Understanding by design*. Vanderbilt University. Center for Teaching. https://cft.vanderbilt.edu/understanding-by-design/

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Caceffo, R., Gama, G., & Azevedo, R. (2018, February). Exploring active learning approaches to computer science classes. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, Baltimore, MD, USA (pp. 922–927).

Chasteen, S., Perkins, K., Beale, P., Pollock, S., & Wieman, C. (2011). A thoughtful approach to instruction: Course transformation for the rest of us. *Journal of College Science Teaching*, *40*(4), 24–30.

Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, *12*(2), ep272. https://doi.org/10.30935/cedtech/8247

Condon, W., Iverson, E. R., Manduca, C. A., Rutz, C., & Willett, G. (2016). *Faculty development and student learning: Assessing the connections*. Indiana University Press.

Eickholt, J. (2018). *Barriers to active learning for computer science faculty. arXiv preprint arXiv:1808.02426*. https://arxiv.org/abs/1808.02426v1

Eickholt, J., Gandy, L., Seeling, P., & Johnson, M. (2019, October). Advancing adoption of active learning pedagogy via new avenues of research and training. In *2019 IEEE Frontiers in Education Conference (FIE)*, Cincinnati, OH, USA (pp. 1–5). IEEE.

Finelli, C. J., Nguyen, K., DeMonbrun, M., Borrego, M., Prince, M., Husman, J., Henderson, C., Shekhar, P., … Waters, C. K.

(2018). Reducing student resistance to active learning: Strategies for instructors. *Journal of College Science Teaching*, *47*(5), 80–91.
Fink, L. D. (2003). *A self-directed guide to designing courses for significant learning*. University of Oklahoma. https://www.bu.edu/sph/files/2011/06/selfdirected1.pdf
Fink, L. D. (2013). *Creating significant learning experiences: An integrated approach to designing college courses*. John Wiley & Sons.
Fraser, S., & Mancl, D. (2021). Engineering for Chaos: Lessons Learned from COVID-19. *ACM SIGSOFT Software Engineering Notes*, *46*(2), 25–27.
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. https://doi.org/10.1073/pnas.1319030111
Gao, J., & Hargis, J. (2010). Promoting technology-assisted active learning in computer science education. *Journal of Effective Teaching*, *10*(2), 81–93.
Giannakos, M. N., Pappas, I. O., Jaccheri, L., & Sampson, D. G. (2017). Understanding student retention in computer science education: The role of environment, gains, barriers, and usefulness. *Education and Information Technologies*, *22*(5), 2365–2382. https://doi.org/10.1007/s10639-016-9538-1
Gill, S. S., Fuller, S., Cabral, A., Chen, Y., & Uhlig, S. (2023). Curriculum Redesign for Cloud Computing to Enhance Social Justice and Intercultural Development in Higher Education. In Meletiadou, E. (Ed.), *Handbook of Research on Fostering Social Justice Through Intercultural and Multilingual Communication* (pp. 62–80). IGI Global.
Gomes, A., & Mendes, A. (2014). A teacher's view about introductory programming teaching and learning: Difficulties, strategies, and motivations. In *IEEE (Ed.)* (pp. 1–7). *Frontiers in Education Conference*. https://doi.org/10.1109/FIE.2014.7044086
Guskey, T. (2020). Flip the script on change: Experience shapes teachers' attitudes and beliefs. *The Learning Professional*, *41*(2), 18–22.
Henderson, C., Beach, A., & Finkelstein, N. (2011). Facilitating change in undergraduate STEM instructional practices: An analytic review of the literature. *Journal of Research in Science Teaching*, *48*(8), 952–984. https://doi.org/10.1002/tea.20439
Hertz, M., & Jump, M. (2013). Trace-based teaching in early programming courses. *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, 91–96. https://doi.org/10.1145/2445196.2445364
Hodges, C. B., Moore, S., Lockee, B. B., Trust, T., & Bond, M. A. (2020). The difference between emergency remote teaching and online learning. *Online Learning*, *24*(1), 6–14. https://doi.org/10.24059/olj.v24i1.2284
Idsardi, R., Hahn, D. A., Bokor, J. R., & Luft, J. A. (2019). Modifying scientific research into introductory science course lessons using a 5E lesson format: An active learning approach. *Journal of College Science Teaching*, *48*(5), 14–21. https://www.jstor.org/stable/26901311
Isaias, P., Miranda, P., & Pifano, S. (2021, July). Learning with the practitioners: Defining and implementing active learning in higher education. In T. Bastiaens (Ed.), *Proceedings of EdMedia + Innovate Learning* (pp. 519–528). Association for the Advancement of Computing in Education (AACE), Waynesville, NC, USA.
Kinzie, J., Landy, K., Sorcinelli, M. D., & Hutchings, P. (2019). Better together: How faculty development and assessment can join forces to improve student learning. *Change: The Magazine of Higher Learning*, *51*(5), 46–54. https://doi.org/10.1080/00091383.2019.1652076
Kovarik, M. L., Robinson, J. K., & Wenzel, T. J. (2022). Why Use Active Learning? In Wenzel, T.J., Kovarik, M.D., & Robinson, J.K. (Eds.), *Active Learning in the Analytical Chemistry Curriculum* (pp. 1–12). American Chemical Society.
Medeiros, R. P., Ramalho, G. L., & Falcao, T. P. (2019). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, *62*(2), 77–90. https://doi.org/10.1109/te.2018.2864133
Mozelius, P. (2022). A Post-Pandemic Strategy to Support Lifelong and Work-Integrated Learning: Rethinking Pedagogy, Technology, and Collaboration. In Connolly, T.M., & Farrier, S. (Eds.), *Leadership and Management Strategies for Creating Agile Universities* (pp. 203–219). IGI Global.
Opfer, V. D., Pedder, D. G., & Lavicza, Z. (2011). The role of teachers' orientation to learning in professional development and change: A national study of teachers in England. *Teaching and Teacher Education*, *27*(2), 443–453. https://doi.org/10.1016/j.tate.2010.09.014
Patton, M. Q. (2014). *Qualitative research & evaluation methods: Integrating theory and practice* (4th ed.). Sage Publications, Inc.
Radermacher, A., Walia, G., & Knudson, D. (2014). Investigating the skill gap between graduating students and industry expectations. In *Companion Proceedings of the 36th International Conference on Software Engineering - ICSE Companion 2014*. https://doi.org/10.1145/2591062.2591159
Sadler, T. D., Foulk, J. A., & Friedrichsen, P. J. (2017). Evolution of a model for socio-scientific issue teaching and learning. *International Journal of Education in Mathematics, Science and Technology*, *5*(2), 75–87. https://doi.org/10.18404/ijemst.55999
Smith, K. (2012). Lessons learnt from literature on the diffusion of innovative learning and teaching practices in higher education. *Innovations in Education and Teaching International*, *49*(2), 173–182. https://doi.org/10.1080/14703297.2012.677599
Thornton, B., Zunino, B., & Beattie, J. (2020). Moving the dial: Improving teacher efficacy to promote instructional change. *Education*, *140*(4), 171–180.
Thota, N. (2014). Programming course design: Phenomenographic approach to learning and teaching. In *2014 International Conference on Teaching and Learning in Computing and Engineering* (pp. 67–74). https://doi.org/10.1109/latice.2014.30
Vihavainen, A., Airaksinen, J., & Watson, C. (2014). A systematic review of approaches for teaching introductory programming and their influence on success. *Proceedings of the Tenth Annual Conference on International Computing Education Research - ICER* 14. https://doi.org/10.1145/2632320.2632349
Watson, C., & Li, F. W. B. (2014). Failure rates in introductory programming revisited. *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education - ITiCSE* 14. https://doi.org/10.1145/2591708.2591749
Wiggins, G. P., & McTighe, J. (2011). *The understanding by design guide to creating high-quality units*. ASCD.
Yin, R. K. (2014). *Case study research: Design and methods* (5th ed.). Sage.

Appendix A. Summer 2020 Understanding by Design Template.

Backward Design: Session 1, Week of May 18, 2020 – Desired Results		
ESTABLISHED GOALS *The enduring understanding and learning goals of the course.*	***Transfer***	
	Students will be able to independently use their learning to… *Refers to how students will transfer the knowledge gained from the course and apply it outside of the context of the course.*	
	Meaning	
	UNDERSTANDINGS *Students will understand that…* *Refers to the **major topics** that students will understand when they complete the course.* *List the major topics for this course / Group the list into **topic categories***	ESSENTIAL QUESTIONS *Refers to the provocative questions that foster inquiry, understanding, and transfer of learning. These questions typically frame the course and are often revisited. If students attain the established goals, they should be able to answer the essential question(s).*
	Acquisition	
	Students will know… *Refers to the **key knowledge, including concepts and vocabulary**, students will acquire from the course.* *Write a statement for each outcome: By the end of the semester, students will demonstrate knowledge of (CS concept) by (concrete and actionable product)*	*Students will be skilled at…* *Refers to the **key skills and applications** students will acquire and practice.* *Write a statement of professional skills that will be learned and practiced in this course. These professional skills may be embedded within outcome statements*

(*Continued*)

(Continued)

Backward Design: Session 2, Week of May 25, 2020 – Evidence and Assessment		
Course Topics	**Assessment Evidence (Products)**	**Knowledge and Professional Skills associated with this product**
Refers to the various course topics that students will be evaluated on.	Decide on the Products students will produce to demonstrate learning associated with each course topic. A product may include one or more student outcomes for the course. Products may be projects, presentations, a set of assignments from the textbook, or in-class assignments. Refers to other types of evidence including quizzes, tests, homework, etc. This is also a good point to consider incorporating selfassessment and student reflections	
Products	**Description of the product**	**Ways to assess the product**
	Describe each product in student-friendly language	*How will you know that the product indicates students know and apply content knowledge and professional skills for each product? Consider using assessments such as peer assessments, rubrics that indicate proficiency descriptions, and products that work versus products that have errors and do not execute*
Backward Design: Session 3, Week of June 1, 2020 – Learning Plan		
This stage encompasses the individual learning activities and instructional strategies that will be employed. This includes lectures, discussions, problem-solving sessions, etc. Use the following information for the planning of each class session.		
Class Preparation		
Readings and learning students need to do to prepare for this lesson (expect students to have completed this work).		
Lesson Information		
Lesson Topic	**Content objectives**	**Professional skill(s) to be learned**
	(Concept, vocabulary, facts, theory, etc.)	
Instruction		
Introduction *(How will you motivate students? Connect to student's background or prior knowledge in an engaging way?)*		

(*Continued*)

(Continued)

Presentation Outline	**Practice/Application**	**Lesson Assessment**
Use as an agenda for the class session (For example, Lecture 10 minutes, activity 15 minutes, lecture 10 minutes again, independent work 10 minutes, 10 peer feedback)	*(Meaningful activities, active learning, interactions between students)*	*(Review content learning objectives and professional skills with students and assess learning)*
Backward Design: Session 3, Week of June 1, 2020 – Learning Plan		
This stage encompasses the individual learning activities and instructional strategies that will be employed. This includes lectures, discussions, problem-solving sessions, etc. Use the following information for the planning of each class session.		
Class Preparation		
Readings and learning students need to do to prepare for this lesson (expect students to have completed this work).		
Lesson Information		
Lesson Topic	**Content objectives**	**Professional skill(s) to be learned**
	(Concept, vocabulary, facts, theory, etc.)	
Instruction		
Introduction *(How will you motivate students? Connect to student's background or prior knowledge in an engaging way?)*		
Presentation Outline	**Practice/Application**	**Lesson Assessment**
Use as an agenda for the class session (For example, Lecture 10 minutes, activity 15 minutes, lecture 10 minutes again, independent work 10 minutes, 10 peer feedback)	*(Meaningful activities, active learning, interactions between students)*	*(Review content learning objectives and professional skills with students and assess learning)*