Blended Approach for Deep Learning: A Framework for Teaching Undergraduate Computer Programming Courses

Mohammad Tafiqur Rahman
Department of Business, Marketing, and Law, University of South-Eastern Norway, Norway

Keywords: Blended Approach, Computer Programming Courses, Deep Learning, Flipped Classroom, Knowledge Construction, University Pedagogy, TEE Approach, Traditional Teaching.

Abstract: Teaching computer programming (CP) courses demands cutting-edge course practicalities that include (i) updated course design with adequate content, (ii) modern pedagogy-enabled course conduction, and (iii) course completion with adequate practically implementable knowledge. However, meeting such requirements is not possible only through the traditional teaching (TT) approach, nor by any specific or individual approaches practiced in modern teaching. We need combined approaches to meet learners' desires and industry needs. I propose a teaching framework that blends traditional and flipped classroom (FC) approaches to facilitate deep learning toward essential knowledge construction on CP and provide practical experiences for software system development. In the proposed framework, the TT approach emphasizes theoretical understanding, whereas the FC approach focuses on active engagement, active participation, and active learning. The TEE (theory-example-exercise) approach binds the chosen approaches together, where the theory part is handled in the TT approach, and the example and exercise parts are processed in the FC approach. Since I successfully applied this blended approach framework to teaching undergraduate CP courses at a Norwegian university, I believe it will be suitable not only for courses in this discipline but also in other disciplines with necessary modifications.

1 INTRODUCTION

Technological advancement has introduced many opportunities to our daily life—from waking up in the morning to going to bed at night, our daily activities are, somehow, affected by digital tools. The education sector is not outside that trend. Rather, it is the most emergent area for digital development (Lundin et al., 2018; O'Flaherty & Phillips, 2015). Exploring and analyzing the technology-rich teaching and learning environment is essential to identifying different challenges and opportunities for designing new courses or upgrading old ones (Divjak et al., 2022). Thus, learning computer programming (CP) is now treated as a requirement to shine in this digital age (O'Flaherty & Phillips, 2015). To facilitate students with smooth teaching and learning experiences in CP courses, we need to incorporate essential technologies into our teaching philosophies along with suitable pedagogy.

Because of its relevance and applicability in real life, learning CP is always on a student's priority list of study (Sambe et al., 2021). We have already experienced how online teaching platforms preserved the education sector during COVID-19 lockdowns (Barr et al., 2020) and are now observing how digitalization is ruling the world during post-pandemic situations. Such technological advancement is not possible without innovative digitalization (Müller et al., 2021), and to do so, essential knowledge of CP is crucial. Hence, programmers are in higher demand in the job market, making current and future students extremely ambitious about their careers (Ouhbi & Pombo, 2020). For proper knowledge construction, they expect to acquire practice-oriented knowledge from the course and gain real-time experience to fulfill their desires (Feijóo-García et al., 2021).

Teachers should inform their students that knowledge is not limited to the two cover pages of the course book or the provided course content; a clear understanding of the topics learned is necessary. Technology-oriented learning can help students gain such an in-depth understanding of the subject (Duggirala et al., 2021). Along with the content from

Rahman, M.
Blended Approach for Deep Learning: A Framework for Teaching Undergraduate Computer Programming Courses.
DOI: 10.5220/0011959900003470
In *Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023) - Volume 2*, pages 394-402
ISBN: 978-989-758-641-5; ISSN: 2184-5026

course books, essential online resources can be incorporated into lectures, assignments, projects, and examinations to stimulate students' critical thinking and train them with the analytical know-how to solve practical problems (Einhorn, 2012). Thus, students may become habituated with self-learning approaches not only for answering examinations creatively but also for solving real-time problems in their careers (Sambe et al., 2021).

Therefore, they require deeper insight and understanding of the course content defined by their teachers with a clear indication of the goal and outcomes of their study and training (Howie & Bangnall, 2015; Paez, 2017). For deep learning, students are expected to be active in learning processes that include understanding the problem and utilizing proper logic and evidence to identify and implement solutions (Entwistle, 2000). They are encouraged to collaborate with their peers and teachers not only to solve the problem but also to evaluate their proposed ideas. Thus, deeper knowledge is constructed (Biggs & Tang, 2011). In this way, deep learning ensures that students have a more comprehensive grasp of the subject being studied and can successfully apply their gained knowledge to the field (Howie & Bangnall, 2013).

To prepare students for their careers, a student-centered, career-focused teaching philosophy needs to be emphasized and applied. Since the traditional way of teaching is mostly lecture-based (Erdogmus & Péraire, 2017), students get little or limited opportunities for discussion, practice, and exploration (Lin, 2021). On the other hand, the student-centric flipped classroom (FC) approach is also limited to provide the abovementioned facilities to students as it requires extra time and effort from both teachers and students (Amresh et al. 2013). For example, in traditional FC approach, teachers are required to prepare and upload video lectures and students need to go through them before attending the session (Elliot, 2014). Thus, to facilitate deep learning for CP students, the student-centric FC concept can be blended with the traditional teaching (TT) approach, as suggested by Divjak et al. (2022) and Gren (2020). But how can such blending be done? Finding a way leads to this study's research question:

How does the blended approach contribute to deep learning and knowledge construction?

In the following section, related works on recently used pedagogical approaches are reviewed, especially for teaching CP courses. The induced framework is presented and described in section 3, and its application and evaluation are discussed in sections 4 and 5, respectively. Finally, section 6 concludes the paper, along with research limitations and future research directions.

2 RELATED WORKS

Academia is no longer just a privileged knowledge provider but fosters a dialogical space to create societal values and human worth (Class et al., 2021). The current academic course design at universities has been highly influenced by social networking, technology, and practice-oriented teaching and learning (Nørgård et al., 2019). Thus, teachers try different approaches before selecting ones that are appropriate to their classes and updating them accordingly.

In recent years, teachers of CP courses have advocated an FC approach (Fetaji et al., 2019), bringing in-class activities out of the classroom (Fulton, 2012). Although widely tried, applying the FC technique in CP is challenging because it covers not only the theory, methods, and tools for developing new informatics solutions (Feijóo-García et al., 2021) but also accepts students from non-scientific backgrounds for admission (Sambe et al., 2021). We are experiencing increased student enrollment in computing education, especially after the COVID-19 pandemic, when we were forced to switch to digital and online platforms (Kawash et al., 2021; Arima et al., 2021).

Although mentioned as an active learning methodology, Olivindo et al. (2021) incorporated the gamification technique within the FC approach to improve students' acceptance and in-class engagement during the COVID-19 pandemic. They reported higher student satisfaction with their adopted approach. Similarly, Lin (2021) combined the learning diagnostic methodology with the FC approach to support students with learning diagnosis activities. According to Lin (2021), simply following the FC approach is not sufficient to provide CP students with adequate learning support or necessary feedback before class. Lin (2021) also reported outstanding performance for students who followed the adopted approach.

El-Glaly (2020) applied the FC concept to her teaching by assigning several related research papers to the class and engaging students through a discussion on selected research papers (three papers per week) and presentations (one student per week). Although the students' and research papers' selection criteria were never discussed, she was supportive of including necessary lectures and providing students with hands-on experience where applicable. Paez (2017) shared his experience with adopting the FC approach in software engineering courses and demonstrated that students are required to maintain a minimum workload during the semester (four to six hours of homework per week for a four-hour long weekly class). This allows for an easy estimation of the dedicated workload for students who take several courses per semester, and each course is instructed using the FC method.

Although Paez (2017) asserts that a FC method–supported teaching approach is suitable for smaller classes with adequate teaching support (two teachers for 15 students), Marasco et al. (2017) found the FC approach helpful in teaching around 800 students with different programming backgrounds enrolled in the first year of their undergraduate study. They extensively utilized the online learning management platform to conduct the introductory CP course by posting weekly video lectures, hosting embedded quizzes, and facilitating student collaboration on course exercises. However, the authors for both articles emphasized, like earlier authors, the extensive redesigning of course practicalities for running courses in the FC format.

According to Barr et al. (2020), laboratory sessions and group work suffer tremendously on online platforms, although CP students engage and learn better while being tied to programming activities (El-Glaly, 2020). Hence, Gren (2020) claims that the FC approach may facilitate students in getting better grades, but a clear understanding of students' perceptions of the course is always missing. Gren suggests a blended learning platform for proper knowledge construction and for its successful establishment. Furthermore, Strayer (2012) recommends enabling or carefully integrating information technology (FC approach, for example) in regular in-class activities (TT approach, for example). Such an approach can provide students with an effective, efficient learning experience; thus, a meaningful subjective linking of the materials they learned in the course can be demonstrated (Garrison & Kanuka, 2004; Strayer, 2012). These meaningful learning experiences point to the deep learning of the concepts/topics covered in the course (Beattie et al., 1997; Howie & Bangnall, 2013).

3 THE FRAMEWORK

Research and discussion on teaching and learning is an ongoing process; the latest ideas and techniques replace old or existing ones for better outcomes. Different approaches focus differently on the two main entities of the process (teacher and student), and their activities, collaboration, and communication. Vaughan et al. (2013) defined blended learning as a pedagogical and technological innovation that is significantly redesigned to enhance students' engagement in the entire learning process. They emphasize bringing both physical and online learning activities into the process rather than simply adding online components. Koehler and Mishra (2009) suggest a proper balance between technology, pedagogy, and content knowledge in course design and conduction. Figure 1 illustrates how the blended approach contributes to students' deep learning and knowledge construction in CP. If sufficiently modified and upgraded, this framework can also be applied to courses in other disciplines.

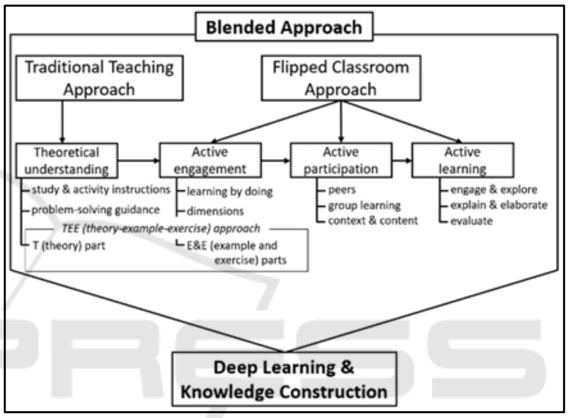

Figure 1: Framework for the blended teaching approach.

The proposed teaching and learning framework in Figure 1 combines the FC approach with the TT approach (hence, conceptualized as a blended approach) (Strayer, 2012). Here, the TT approach works to develop an adequate theoretical understanding of the course content (as suggested by McNally et al. (2017)), whereas the FC approach focuses on confirming students' active engagement, participation, and learning. This combination of entities constructs deeper knowledge of the course content. The overall knowledge construction process is elaborated on in the rest of this section.

To actuate the blending, I introduced the theory-example-exercise (TEE) approach to deliver lectures for CP courses by following a static order of three steps: (i) theory or concept, (ii) coding examples, and (iii) testing exercises. In this approach, the teacher starts each class with a regular lecture in the traditional way. This includes concept development, providing study materials, and offering problem-solving guidance with sufficient activity instructions (Kim et al., 2014). As soon as theory building is confirmed, the teacher provides related examples (practice coding) for deeper understanding of the topic/concept and for preparing students for problem-solving (exercises) sessions. This unobstructed vision of course conduction is essential in the FC approach because it protects students from being disengaged with the course and its designed activities (Strayer, 2012). To facilitate quick access to the course content, the teacher must share all the covered materials (lecture, examples, exercises, homework, etc.) on the online platform (e.g., Canvas, Learning Management System, GitHub) before or immediately after each session.

Practice coding can be conducted in three ways: (i) coding together: coding is done by the teacher and the students follow him/her; (ii) supervised coding: the code is provided by the teacher in a non-copyable format (e.g., jpeg, png, gif), and the students type the code to see the result; and (iii) combined coding: mixing (i) and (ii). While coding, it is important that the teacher explains the code in every viable way. Hence, the teacher should bring smaller and simpler problems for the "example" session, whereas extensive problems can be saved for the "exercise" session to test students' understanding.

In problem-solving (or "exercise") sessions, the teacher sets complex and extensive programming exercises that the students are required to solve in class. They are encouraged to code the assigned exercises by themselves so that they can gain a good foundation in problem solving. All kinds of discussion, brainstorming, analysis, and coding must be done in groups, and the teacher provides adequate support and guidance in person for solving the programming problems. Thus, the TEE approach ensures that students successfully practice the "learning by doing" method, which is the absolute learning technique for CP courses (Kawash et al., 2021). Furthermore, it can also be a regular practice to recap the main point of the previous lecture at the beginning of the class so that the students receive the opportunity to clear their conceptual misunderstandings, if any.

Thus, the TEE approach influences active engagement and its components as proposed by Fredricks et al. (2004)—behavioral, cognitive, and emotional/affective. It ensures students' involvement in active learning by responding to the teacher's directions in activities (behavioral engagement), preference in problem-solving activities and fault tolerance (cognitive engagement), and commitment to belonging and values in group activities (emotional engagement). Since the FC method emphasizes students' active engagement in the learning process, it contributes well to deep learning (Jensen et al., 2015; Steen-Utheim & Foldnes, 2018).

To ensure active learning, the FC approach requires proper activity design. Bybee et al. (2015) proposed a 5E instructional model for improved active learning (applied in Jensen et al. (2015)): engage, explore, explain, elaborate/extend, and evaluate. Hence, in conjunction with in-class activities, assignments and projects should be designed in the context of inquiry-based learning that incorporates role plays, simulations, brainstorming, and so on. To enrich their software engineering understanding of system development, students are encouraged to work with real-time clients to build their projects on practical scenarios. Student groups should be able to explain and elaborate on their projects and divide them into realistic, meaningful, and achievable milestones. Thus, they can make hands-on observations, solve interesting and practical problems, and achieve explorable models of practical projects in real life. Such an accomplishment makes them ready to plan and carry out successful computer system development projects together with others in practical setups. Finally, various initiatives can be taken to evaluate students' learning, both individually and in groups. For example, passing group assignments/project can be set as a prerequisite for the individual final examination. Thus, both their deep learning (conceptual and analytical capabilities) and their surface learning (memorizing capabilities) can be effectively measured and correctly valued (Photopoulos et al., 2021).

Altogether, the proposed blended teaching and learning approach is expected to help students sharpen their skills not only for developing dynamic full-fledged software applications but also for planning and carrying out tasks from scratch (as projects) with others. They can utilize their competencies in project management, system development, and programming to pave a smoother software engineering journey. In this way, their professional qualifications (i.e., knowledge construction) can be highlighted for their future careers in practice (i.e., deep learning) (Bachnak & Maldonado, 2014).

4 DISCUSSION

The FC concept expects students to cover literature and short video lectures before coming to class and applying their understanding to classroom activities (El-Glaly, 2020; Lin, 2021). However, this technique is not fully effective for CP courses because students may struggle to understand complex programming concepts, business logic, and mathematical algorithms. In my teaching, I experienced my students struggling to understand a concept by watching the short video lectures posted either by me (as their teacher) or on other online platforms. The situation was worse for weaker students, who often expect teachers to go through complex logic and algorithms in person in class, even when study materials are made available before lectures. Since adequate and in-time learning is the focal point in academia, educators cannot overlook the demands or desires of students. This indicates doubling the time used for the same tasks (Phillips & Trainor, 2014). The time both parties already invested becomes useless—the teacher must conduct the lecture once again, and so the students attend it twice. We can avoid this inconvenience by starting the class with traditional lectures and then gradually incorporating the FC approach; thus, the blended approach can easily address this problem.

Conducting lectures (for theory) and laboratory sessions (for practice coding) in separate slots is a regular practice for teaching CP courses. Such a pairing sometimes happens in back-to-back slots or weekly slots (Marasco et al., 2017). Again, this is ineffective (to some degree) since students get involved in many other activities during the gap between theory and laboratory sessions, and thus may forget some parts of the learned concept they are expected to apply in laboratory sessions. To address this issue, Bachnak and Maldonado (2014) emphasized students' intensive involvement in their education and the application of their learning. The TEE approach helps facilitate such extensive learning by conducting laboratory sessions in parallel with lectures. However, practicing this approach is not possible for a shorter class time; instead, a four-hour-long weekly class can be conducted with sufficient breaks in between. The success ratio of this practice is demonstrated in the next section, along with student feedback.

Making students code in class is another challenge for teaching CP courses. Students' expectations in such courses vary to a large extent: One group of students may have high expectations of coding in class, while another group may prefer coding later or at home. Better students may grab the concept easily and quickly, whereas weaker students may struggle. Thus, it becomes difficult for teachers to balance class activities with a moderated workload, because better students may easily get bored if the pace is too slow and the weaker students may struggle to up if the pace is too fast. Alternatively, students may feel insecure about coding flexibility if the "coding together" method is followed. Slow typing is time-consuming, and erroneous code could demand the entire class time to fix a bug that could go unsolvable. For example, missing a simple semicolon (;) is enough to ruin the productivity of the class. Thus, a class would generate a small outcome in which the expectation was high. Therefore, to keep students interested and engaged in classroom activities, the TEE approach encourages teachers to prepare exercises with extended complexities. The brighter students can attempt to solve these problems by utilizing their capabilities of applying advanced logic.

However, Ouhbi and Pombo (2020) highlight that enhancing students' class participation is the greatest challenge for teaching CP—they might not participate in class discussions, answer questions, take the lead in group activities, and so on. Students rarely ask questions to their teachers during class time or mostly feel insecure about answering the questions asked (Strayer, 2012). They foster a mindset of being teased by fellow classmates if they answer incorrectly. However, there are always a couple of familiar faces who answer questions or discuss raised issues. To ascertain students' understandings and guide their knowledge construction, the proposed blended approach suggests that teachers speak to individual students in person and monitor their activities on their computer screens. Thus, they could step out of their cocoons, although some students may find it inconvenient due to their screen privacy.

Additionally, active engagement does not bring about active participation all the time (Strayer, 2012). Free riders are always there; they try to escape hurdles but enjoy group grading. For example, some members engaged in group activities (e.g., discussion, task analysis, requirement elicitation, etc.) but did not participate enough in coding for software system development. Such piggybacking brings overhead to other group members and thus, in most cases, produces poor group performance. Hence, to make the proposed blended approach functional toward in-depth knowledge construction, forming effective groups is one of the key requirements (Gren, 2020), and it should not be done randomly (Barr et al., 2020). It is important that teachers guide group members to encompass a good understanding not only of the working attitude and responsibilities but also of communication, information sharing, and leadership (Oliveira & da Silva Borges, 2021). Therefore, students should be independent to form their own groups and select their group leaders by themselves. This will help them become accustomed to co-learning in various contexts and complete the course project successfully. To stop free-riders, groups must be restricted to a manageable size (four to six students per group) (Barr et al., 2020; Gren, 2020). It is true that self-formulating group establishment is rarely practiced in real-time setups; however, we can use it in academia for better student engagement in assigned tasks/projects and active participation in group learning.

Effective teaching and learning require both teachers and students to actively participate in the learning process. Teachers are authority figures in this pedagogy model who provide students with essential study materials and supervise their learning (Steen-Utheim & Foldnes, 2018). Thus, to meet the curriculum objectives, the proposed blended approach clarifies the expectations, stipulates the objectives, and assigns the required activities to students at the beginning of each class. According to Brookfield (2017), these are the requirements for generating effective teaching methods. In addition to the planned classroom lectures, CP teaching pedagogy should include in-class problem solving, self-learning, relevant classroom entertainment, question-and-answer sessions, presentation and demonstration sessions, and support for technical report writing. Besides organized instructions on physical and/or digital lectures, the necessary guidance and supervision are required to be provided to groups and individuals on case study discussions, group work, assignments, and projects (Kim et al., 2014). To strengthen understanding of the topics covered, the blended teaching approach encourages students to use online video lectures upon necessity. Since teaching and learning are connected, this approach suggests a continuous assessment of student learning during the semester.

5 FRAMEWORK EVALUATION

Following Bachnak and Maldonado's (2014) recommendations, student feedback on course structure and conduction was used to evaluate the proposed framework. A survey was designed using Google Forms and sent to 70 students who completed an introductory web programming course in the 2021 fall semester at a Norwegian university. In the survey, the participants were asked to answer two questions ("How did you find the content for this course?" and "How did you find the lecturing in this course?"), and to provide their reflections as free-text comments. Analyzing these free-text comments can guide quality improvement initiatives, supported by deeper insights from students' experiences (Arditi et al., 2020). By anonymously evaluating the course, 42 students evaluated the proposed framework for the blended teaching and learning method. The analytical results are presented below. To restrict students from responding neutrally, a Likert scale of 6 was used in this survey.

A summary of the students' written feedback (free-text comments) is presented in Table 1. It demonstrates not only the strengths and weaknesses of the course and the teaching they emphasized, but also their recommendations for future improvement in the course.

Table 1: Students' reactions to course content and teaching.

Strength		weaknesses		Recommendation	
Topic	Frequency	Topic	Frequency	Topic	Frequency
Course Structure					
Rating (out of 42): 1(1), 2(5), 3(2), 4(17), 5(11), 6(6): (19% + 81%)					
Coding	20	No noticeable weakness	15	More exercise & example	15
Course structure	6	Class & home exercise	6	Coding together	8
Course conducting style	5	Coding (JS+HTML)	5	The course is well formed	4
Course interest	4	Course structure & conduction	4	Guest Lecture (no/less)	2
Course conducting approach	3	Theory (HCI)	4	Group meeting with lecturer	1
Canvas & classwork	2	Assignments	2	Speed	1
		Guest Lecture	1		
		Speed	1		
		Slides	1		
Teaching					
Rating (out of 42): 1(1), 2(6), 3(4), 4(15), 5(9), 6(7): (26% + 74%)					
Teaching approach	9	No noticeable weaknesses	13	Keep going as you are	6
Explanation & answering qu.	8	Coding together	5	Coding together	5
Examples & exercises	7	Language	4	More exercise/examples	3
Lecturing & interaction	6	Explanation & answering que.	3	Less theory	2
Coding	5	Voice & talking to the point	3	Specific outcomes	2
Canvas	1	Teaching approach	2	Language & voice	2
Slides	1	Speed	2	Guest Lecture	1
Everything	1	Guest Lecture	2	Using technical words	1
		Theory	2	Explanation	1
		Exercises	1	Course syllabus	1
		Sound system	1		
		Using technical words	1		
		Slides	1		
		Less discussion on assignments	1		

Most of the students who completed the survey were happy with the course content and lecturing. Although "no noticeable weakness" was mostly reported, it can be identified that they expected more examples and exercises in both formats (in-class and homework), which is good. They preferred the "coding together" method rather than the implemented "supervised coding" method. However, they were not happy with the amount of content covered in the class—they wanted less. Students looked for direct answers to their questions without getting a heavy background and not using technical words. Although they reported good reviews for course conducting style, lecturing, and interaction, they suggested further development for teaching language and lecturing slide content (some of them preferred Norwegian speakers and more content in the slides). Lastly, students expected more discussion on the course structure, assignments, evaluation, and conduction.

The feedback on the course content and organization was positive. More than 80% of the students found that the course topics improved their software system development skills. They liked the lectures, assignments, and literature because they could connect the theory they learned in the course to real-world web application development. Figure 2 demonstrates the statistics.

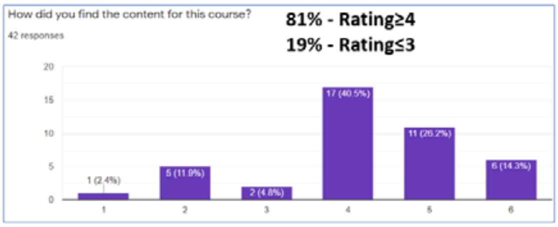

Figure 2: How much the students liked the course.

To understand and improve teaching quality, students were also requested to provide their feedback on lecturing. 74% of students found the implemented TEE approach was helpful to them in following the course content. Besides this online survey, they were invited to have discussions on course structure, content, and conduction in person and to provide feedback accordingly. Instead of dividing the class between theory and laboratory sessions, the TEE practice was well accepted in the class. They appreciated the immediate help they received from the teacher when stuck somewhere in the code. Figure 3 demonstrates the statistics.

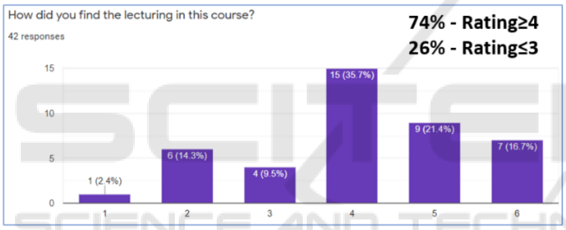

Figure 3: How much the students liked the teaching.

Altogether, the students' feedback helped me, as the course teacher, identify the strengths and weaknesses of the course design and its conduction. By analyzing their reflection data, we can easily identify the concentration points for further development of the course. Students sometimes suggest or recommend their preferences and thus contribute to improving the course and lecturing to achieve balanced teaching and learning. Hence, such an evaluation of the overall teaching framework—course content, assessment forms, course organization, and teaching activities—is always considered a resource for teachers of different courses. It will assist us in improving our teaching in the next semester and to be more prepared to master the course in the case of a sudden shift between offline and online modes of teaching.

6 CONCLUSIONS

This study presented and evaluated a framework for a blended approach to teaching CP, such as in technical courses. It discussed how this teaching method enables students to gain an adequate understanding of the subject matter and apply it in practice. To provide deep learning for proper knowledge construction, this pedagogical technique merges traditional teaching with the flipped classroom approach. It used the TEE approach to perform this merging, where the theory was addressed in traditional lectures to understand the study foundation, and example and exercise parts were used to ensure students' active engagement in their education as well as their active participation and learning in the flipped classroom. This overall approach was found to be effective by the students who participated in the course. Like any other research, this study has some limitations. The proposed teaching technique was applied to a specific first-semester course, where basic programming was taught. Its applicability to other advanced programming courses and its acceptance to other faculty members remains unevaluated. For future work, I intend to investigate students' recommendations for their applicability in further adjustment of the proposed framework and enhance it accordingly by incorporating a structured process for course design and course evaluation, especially the examination system for computer programming courses.

REFERENCES

Amresh, A., Carberry, A. R., & Femiani, J. (2013). Evaluating the effectiveness of flipped classrooms for teaching CS1. In *FIE* (pp. 733–735). IEEE.

Arditi, C., Walther, D., Gilles, I., Lesage, S., Griesser, A. C., Bienvenu, C., Eicher, M., & Peytremann-Bridevaux, I. (2020). Computer-assisted textual analysis of free-text comments in the Swiss Cancer Patient Experiences (SCAPE) survey. *BMC Health Service Research*, *20*(1), 1–12.

Arima, S., Assilmia, F., Maekawa, M. S., & Okawa, K. (2021). Design and practice of ESD in high school in Japan through online video co-creation workshop. In *CSEDU (1)* (pp. 640–647). SciTePress.

Bachnak, R., & Maldonado, S. C. (2014). A flipped classroom experience: Approach and lessons learned. In *ASEE* (pp. 24–50). ASEE PEER.

Barr, M., Nabir, S. W., & Somerville, D. (2020). Online delivery of intensive software engineering education during the COVID-19 pandemic. In *CSEE&T* (pp. 1–6). IEEE.

Beattie, V., Collins, B., & McInnes, B. (1997). Deep and surface learning: A simple or simplistic dichotomy? *Accounting Education*, *6*(1), 1–12.
Biggs, J., & Tang, C. (2011). *Teaching for quality learning at university*. McGraw-Hill Education (UK).
Brookfield, S. D. (2017). *Becoming a critically reflective teacher*. John Wiley & Sons.
Bybee, R. W., Taylor, J. A., Gardner, A., Van Scotter, P., Powell, J. C., Westbrook, A., & Landes, N. (2006). The BSCS 5E instructional model: Origins and effectiveness. *Colorado Springs, Co: BSCS*, *5*, 88–98.
Class, B., Soulikhan, F., Favre, S., & Cheikhrouhou, N. (2021). A framework for an open education supply chain network. In *CSEDU* (pp. 617–625). SciTePress.
Divjak, B., Rienties, B., Iniesto, F., Vondra, P., & Žižak, M. (2022). Flipped classrooms in higher education during the COVID-19 pandemic: Findings and future research recommendations. *IJETHE*, *19*(1), 1–24.
Duggirala, V. D., Butler, R. S., & Kashani, F. B. (2021). iTA: A digital teaching assistant. In *CSEDU (2)* (pp. 274–281). SciTePress.
Einhorn, S. (2012). Microworlds, computational thinking, and 21st century learning. *LCSI White Paper*, 1–10.
El-Glaly, Y. N. (2020). Teaching accessibility to software engineering students. In *TSCSE* (pp. 121–127). IEEE.
Elliot, R. (2014). Do students like the flipped classroom? An investigation of student reaction to a flipped undergraduate IT course. In *FIE* (pp. 1-7). IEEE.
Entwistle, N. (2000). Promoting deep learning through teaching and assessment: Conceptual frameworks and educational contexts. In *TLRP (1)*, (pp. 9-20). ESRC.
Erdogmus, H., & Péraire, C. (2017). Flipping a graduate-level software engineering foundations course. In *ICSE: Software Engineering Education and Training Track (ICSE-SEET)* (pp. 23–32). IEEE.
Feijóo-García, M. A., Ramírez-Arévalo, H. H., & García, P. G. F. (2021). Collaborative strategy for software engineering courses at a South American university. In *CSEDU (2)* (pp. 266–273). SciTePress.
Fetaji, M., Fetaji, B., & Ebibi, M. (2019). Analyses of possibilities of flipped classroom in teaching computer science courses. In *MIPRO* (pp. 747–752). IEEE.
Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School engagement: Potential of the concept, state of the evidence. *Review of Education Research*, *74*(1), 59–109.
Fulton, K. (2012). Upside down and inside out: Flip your classroom to improve student learning. *Learning & Leading with Technology*, *39*(8), 12–17.
Garrison, D. R., & Kanuka, H. (2004). Blended learning: Uncovering its transformative potential in higher education. *JIHEDUC*, *7*(2), 95–105.
Gren, L. (2020). A flipped classroom approach to teaching empirical software engineering. *IEEE Transactions on Education*, *63*(3), 155–163.
Howie, P., & Bagnall, R. (2013). A critique of the deep and surface approaches to learning model. *Teaching in Higher Education*, *18*(4), 389–400.
Howie, P., & Bagnall, R. (2015). A critical comparison of transformation and deep approach theories of learning. *IJLE*, *34*(3), 348–365.
Jensen, J. L., Kummer, T. A., & Godoy, P. D. D. M. (2015). Improvements from a flipped classroom may simply be the fruits of active learning. *CBE—LSE*, *14*(1), ar5.
Kawash, J., Horacsek, J., & Wong, N. (2021). What we learned from the abrupt switch to online teaching due to the COVID-19 pandemic in a post-secondary computer science program. In *CSEDU* (pp.148–155). SciTePress.
Kim, M. K., Kim, S. M., Khera, O., & Getman, J. (2014). The experience of three flipped classrooms in an urban university: An exploration of design principles. *JIHEDUC*, *22*, 37–50.
Koehler, M., & Mishra, P. (2009). What is technological pedagogical content knowledge (TPACK)? *JCITTE*, *9*(1), 60–70.
Lin, Y. T. (2021). Effects of flipped learning approaches on students' learning performance in software engineering education. *Sustainability*, *13*(17), 9849.
Lundin, M., Bergviken Rensfeldt, A., Hillman, T., Lantz-Andersson, A., & Peterson, L. (2018). Higher education dominance and siloed knowledge: A systematic review of flipped classroom research. *IJETHE*, *15*(1), 1–30.
Marasco, E. A., Moshirpour, M., & Moussavi, M. (2017). Flipping the foundation: A multi-year flipped classroom study for a large-scale introductory programming course. In *ASEE (pp. 1-9)*. ASEE PEER.
McNally, B., Chipperfield, J., Dorsett, P., Del Fabbro, L., Frommolt, V., Goetz, S., Lewohl, J., Molineux, M., Pearson, A., Reddan, G., Roiko, A, & Rung, A. (2017). Flipped classroom experiences: Student preferences and flip strategy in a higher education context. *Higher Education*, *73*(2), 281–298.
Müller, W., Grassinger, R., Schnebel, S., Stratmann, J., Weitzel, H., Aumann, A., Bernhard, G., Gaidetzka, M., Heiberger, L., Kreyer, I., Schmidt, C., Uhl, P., Visotschnig, M.S., & Widmann, J. (2021). Integration of digital competences into a teacher education program: A sensitive approach. In *CSEDU (1)* (pp. 232–242). SciTePress.
Nørgård, R. T., Mor, Y., & Bengtsen, S. S. (2019). Networked learning in, for, and with the world. In *NPL* (pp. 71–88). Springer, Cham.
O'Flaherty, J., & Phillips, C. (2015). The use of flipped classrooms in higher education: A scoping review. *The Internet and Higher Education*, *25*, 85–95.
Oliveira, E. W., & da Silva Borges, M. R. (2021). Supporting educators to design collaborative learning scenarios. In *CSEDU (2)* (pp. 95–103). SciTePress.
Olivindo, M., Veras, N., Viana, W., Cortés, M., & Rocha, L. (2021). Gamifying flipped classes: An experience report in software engineering remote teaching. In *BSSE* (pp. 143–152). ACM.
Ouhbi, S., & Pombo, N. (2020). Software engineering education: Challenges and perspectives. In *EDUCON* (pp. 202–209). IEEE.
Paez, N. M. (2017). A flipped classroom experience teaching software engineering. In *SECM* (pp. 16–20). IEEE.

Phillips, C. R., & Trainor, J. E. (2014). Millennial students and the flipped classroom. *ASBBS*, *21*(1), 519.

Photopoulos, P., Tsonos, C., Stavrakas, I., & Triantis, D. (2021). Preference for multiple choice and constructed response exams for engineering students with and without learning difficulties. In *CSEDU (1)* (pp. 220–231). SciTePress.

Sambe, G., Drame, K., & Basse, A. (2021). Towards a framework to scaffold problem-solving skills in learning computer programming. In *CSEDU (1)* (pp. 323–330). SciTePress.

Steen-Utheim, A. T., & Foldnes, N. (2018). A qualitative investigation of student engagement in a flipped classroom. *Teaching in Higher Education*, *23*(3), 307–324.

Strayer, J. F. (2012). How learning in an inverted classroom influences cooperation, innovation and task orientation. *JLER*, *15*(2), 171–193.

Vaughan, N. D., Cleveland-Innes, M., & Garrison, D. R. (2013). *Teaching in blended learning environments: Creating and sustaining communities of inquiry*. Athabasca University Press.