Education and Information Technologies (2023) 28:5673–5695
https://doi.org/10.1007/s10639-022-11328-6

Emerging from the pandemic: instructor reflections and students' perceptions on an introductory programming course in blended learning

Thitima Srivatanakul[1]

Received: 30 May 2022 / Accepted: 4 September 2022 / Published online: 4 November 2022

Abstract

Teaching an introductory programming course to first-year students has long been a challenge for many college instructors. The COVID-19 pandemic, which caused unprecedented shifts in learning modality across the globe, has worsened the learning experience of novice programmers. Instructors have to find innovative ways to keep students engaged and learning. Blended or hybrid learning has become a new preferred way of learning during the COVID-19 pandemic. Blended learning is viewed as a combination of both in-person and online instructions. Such a learning environment offers instructors the flexibility to provide learners with an engaging face-to-face learning experience while promoting the well-being and safety of students. Starting Fall 2020, York College (and other CUNY colleges) has since offered several courses in hybrid mode. Two years have passed since the abrupt transition. There were several lessons learned from the experiences. In this paper, I discussed evidence-based pedagogical approaches that were used to teach students in an introductory computer programming class at York College, CUNY, where blended learning was used. Student perceptions of learning experience and obtaining coding skills in both online and in-person environments are also presented. The findings from the survey suggested that students benefited from face-to-face interactions and feedback, while those who preferred an online environment liked the flexibility that online components offer. Through careful design and implementation of pedagogical approaches used in the class, novice programmers could potentially benefit from both face-to-face and online components of blended learning.

Keywords Introductory programming · Blended learning · Hybrid mode · COVID-19 · Student perceptions

✉ Thitima Srivatanakul
tsrivatanakul@york.cuny.edu

[1] York College of The City University of New York, 94-20 Guy R. Brewer Boulevard, 11451 Jamaica, NY, United States

1 Introduction

The COVID-19 pandemic in 2020 has brought dramatic and unprecedented shifts in learning modality to schools and colleges around the world. K-12 schools in the USA have adopted remote learning as part of their instructions for over a year (Zota & Granovskiy, 2021). Universities also faced a major shift from face-to-face instruction to remote learning when the pandemic struck. The instruction mode has been forced to change from an in-person setting to a distance learning for the remaining portion of the classes in Spring 2020 in most universities, including the City University of New York (CUNY, 2020). This unanticipated change posed several challenges to the course instructors and students. It was only until the 2021–2022 academic year that we started to see relative normalcy. For example, public schools in New York City are no longer providing remote learning options to students starting Fall 2021. CUNY campuses are gradually returning to primarily in-person instructions. Nevertheless, many instructors and students have had a rough period adjusting to the new learning modality, technologies, and environment.

Two years have passed since the abrupt transition. There were several lessons learned from the remote learning experiences (Vollbrecht et al., 2020; Toti & Alipour, 2021; Capdevila, 2021). Although online classes offer more flexibility, students find completing tasks and interacting with their instructors more challenging (Toti & Alipour, 2021). Delivering course materials in synchronous online instruction takes longer online than in-person (Vollbrecht et al., 2020). Courses that require laboratory and hands-on activities, like programming courses, pose more challenges in a remote learning environment (Capdevila, 2021). Specifically, the shift to online learning has changed how students learn to acquire coding skills in the classroom.

Nonetheless, various tips and advice were shared with the aim of improving student learning and engagement. The cognitive empathy activities used in (Capdevila, 2021) proved to be an effective means to improve student performance. Building in assessments into live online instruction, such as polling, allows instructors to check student understanding before proceeding to the next topics (Vollbrecht et al., 2020). Specifically, different teaching strategies during the pandemic were discussed in computer programming courses (Tian, 2021; Hamid and Rashid, 2020; Seeling, 2020). Since we are emerging from the global pandemic, it is now a good time to reflect on the experience and lessons learned. Throughout the experience in teaching an introductory programming course for the past two years during the pandemic, the author has applied several evidence-based pedagogical approaches, which are believed to help alleviate the problems with student learning and engagement in an online environment.

In this paper, I will discuss the pedagogical approaches that were used to teach students in an introductory computer programming class at York College, CUNY, where blended learning (a combination of face-to-face and online components) was used in the Spring 2022 semester. Different instructional approaches are highlighted to contrast how they are implemented in both in-person and online classes. Student perceptions of learning coding skills via an online and an in-person environment are also presented.

2 Background

2.1 Introductory programming courses

At an undergraduate level, an introductory programming course (IPC), or often known as 'CS1', aims to teach students with fundamentals of programming in computing and general problem-solving methods using a high-level programming language such as C++, Java, or Python. The course typically serves as a prerequisite for other computer science advanced courses in CS or related majors. Such a course could appear under different course names, such as "Introduction to Programming", "Introduction to Computer Programming", "Introduction to Computer Science", "Introduction to Computing", or "Computer Science I" at York College, CUNY. This course would be the first programming course for students with no prior exposure to a programming course in their high school years.

Teaching an introductory programming course to first-year students is a challenge for many college instructors (Alammary, 2019). High failure and drop-out rates in introductory programming courses are common. There are several contributing factors to the problem. As stated in a study by (Robins et al., 2003), learning to program involves complex and variety of cognitive activities. Students must acquire new knowledge (e.g., know how a 'for' loop works), related programming strategies (e.g., know how to use and apply a 'for' loop to solve a problem in a program) and have opportunities to practice the skills (Robins et al., 2003). In addition, learning to program entails various other skill sets, such as critical thinking and problem-solving skills, which most novice learners find difficult to acquire (Mehmood et al., 2020). In addition, a study by Malik et al., (2022) stated that the emphasis had been given to programming knowledge (syntax and semantics), and less attention was provided to problem-solving skills in introductory programming courses and textbooks. Other programming-related skills identified in (Medeiros et al., 2018) include mathematical ability, abstraction, and previous knowledge of programming. General education skills include basic knowledge of English, critical thinking and discussion skills, creativity, as well as time management (Medeiros et al., 2018). The lack of mastery of these skills could potentially make it more challenging for students to learn programming.

There is a growing interest by researchers and educators in trying to understand challenges faced by novice programmers in an introductory programming course. Work by Cherenkova et al. (2014) investigated coding problems that students have trouble with. Their work identified that applications of loops are found to be particularly problematic throughout the course. The use of control structures and data structures was also identified as a problem for novice programmers (Yamamoto et al., 2011; Xinogalos 2016). Several strategies should be employed when teaching novice learners such a complex concept. There is an increasing trend in work done in this area. In teaching introductory programming courses, work by Rahman et al. (2019) showed that an interactive and collaborative learning environment yields positive outcomes for students' motivation, engagement, and learning experience. Timely and targeted feedback is important to address learners' weaknesses and gaps (Omer et al., 2021). There are numerous studies that explored teaching tools, pedagogi-

cal approaches, and student learning in introductory programming courses (Luxton-Reilly et al., 2018). For example, meaningful and timely feedback can help students build their confidence and improve enthusiasm for learning to code. Collaboration, such as pair programming and team-based learning, can help to improve students' motivation, engagement, and understanding of the new concepts.

Over the past few years (2018–2021), several review papers on introductory programming courses have emerged. Mehmood et al., (2020) analyzed and synthesized the existing literature to understand the trends in the field of introductory programming course research. Their findings, from nearly sixty research articles, concluded that curricula contents, assessment tool/design, and teaching/learning through tools had not been addressed as much by the majority of the studies (Mehmood et al., 2020). A systematic literature review by Medeiros et al., (2018) aims to gain a better understanding of introductory programming problems. Their work provides a comprehensive review of a hundred papers on students' background knowledge and skills and challenges faced by both the students and instructors. As a result of their findings, it is worth noting that the student motivation and engagement aspect was addressed the most frequently by the studies. Methods and tools for teaching programming were also discussed as one of the faculty challenges in the majority of the papers. Luxton-Reilly et al. (2018) summarized the developments in research and practice in the introductory programming literature over the past 15 years. This study examined research across the breadth of introductory programming, categorizing the work into studies that are relevant to students, teaching, curriculum, and assessment. Their work identified that there is a growth in the number of publications on introductory programming education, especially in the area of tools that support teaching and learning and techniques and activities used.

A recent work by Omer et al., (2021) has conducted the latest developments in the field of introductory programming. The work summarized the major findings of each article reviewed. This provides other researchers and educators with a very comprehensive resource as to which teaching-focused areas could help improve students' performance and motivation, for example. Another main contribution of their work is the proposed taxonomy of introductory programming course aspects and the advice on the effective practices for instructors.

2.2 Teaching introductory programming course in blended Learning Environment

As defined in Graham (2006), "blended learning systems combine face-to-face instruction with computer-mediated instruction". Traditionally, face-to-face instruction occurs live synchronously in a teacher-directed environment, while computer-mediated instruction typically occurs in a remote and asynchronous learning environment (Graham, 2006). As accepted by most scholars, blended learning is usually referred to as a combination of online and face-to-face instruction, where online instruction is not only restricted to asynchronous learning. Blended learning is also referred to as hybrid learning or mixed learning (Tayebinik & Puteh, 2013). Alammary et al. (2017) identified five different blended learning components, namely (1) face-to-face instructor-led, (2) face-to-face collaboration, (3) online instructor-led, (4)

online collaboration, and (5) online-self paced. Thus, in this work, blended learning is viewed as a combination of the face-to-face (or in-person) and online components.

The advantages of applying blended instruction have been explored by several studies. For example, blended learning supports students with different learning styles since they can learn and access resources in a variety of ways (Kaur, 2013). Blended learning provides both instructors and students the flexibility in learning while still benefiting from face-to-face support and instruction.

In introductory programming courses, instructors have also applied blended learning approaches in their classes. A systematic review study by Alammary (2019) reflects how important blended learning is to introductory programming courses. The work synthesized thirty-eight studies that have applied blended learning in introductory programming courses. Here is a list of blended learning models proposed by Alammary (2019) of how the combination of different blended learning components can be applied.

The findings from Alammary (2019) identified that "blended learning has a positive effect on teaching, with students also identifying that blended courses effectively support learning". A balanced and meaningful mix of the different components in introductory programming courses can bring advantages from both online and face-to-face components (Alammary, 2019). Face-to-face formats in introductory programming courses provide students with closer and direct interaction and communication with their instructors and peers. They can get direct feedback and help when needed. Online components enable students in introductory programming courses to manage their own learning plans with greater flexibility. The online environment also provides better access to resources and learning materials used in the course. The work also highlighted that the 'mixed model', where content delivery and practical coding activities occur both face-to-face and online, would potentially be most appropriate for introductory programming courses over other models. Some of the examples of a mixed model include:

Table 1 Blended learning models proposed by Alammary (2019)

Model	Description
Flipped Model	Programming concepts explained using online resources. Active learning focusing on coding and problem solving takes place in a face-to-face environment.
Mixed Model	Course content and practical coding activities take place both face-to-face and online.
Flex Model	Course content and practical coding activities take place online, with face-to-face sessions for feedback and/or evaluation.
Supplemental Model	Course content and practical coding activities take place face-to-face, with online activities available as supplemental.
Online-practicing Model	Course content is delivered through face-to-face and/or online, but an online programming environment is the core for students to practice coding.

- Half of the course is traditional face-to-face lectures, while the other half takes place online.
- Course content is delivered online, but difficult concepts take place in a face-to-face environment.
- Course content is delivered online, but exercises, labs, collaborative activities, or opportunities for questions take place in a face-to-face environment.

2.3 Challenges of introductory programming courses in the COVID era

A number of studies have investigated the impacts of the COVID-19 pandemics on students in introductory programming classes. YeckehZaare et al. (2022) concluded in their study that the COVID-19 negatively impacts the quantity and quality of studying of students in an introductory computer science course during the pandemic. This was measured by comparing the number of days and eBook interactions of students prior to and during the pandemic. Lohiniva and Isomöttönen (2021) explored novice programming students' motivation during the pandemic. Their work concluded that the pandemics influenced the students' study motivation regardless of the modality (online or in-person). Difficulties in communication, collaboration, and commitment were among the contributing factors. To increase study motivation, students should have good support from their instructors and peers, as well as the opportunity to engage and collaborate with others (Lohiniva and Isomöttönen, 2021).

Lee et al. (2021) discussed how they assessed the programming skills of students in an introductory Python programming course when the final exam could not be conducted on campus. The in-person final exam was replaced with an open-ended game assignment followed by a virtual oral exam where each student had to explain the work submitted individually. Tian (2021) discussed a case study of teaching an introductory programming course in a hybrid mode during the pandemic. The survey conducted in their study suggested that students preferred a traditional classroom setting over remote learning to learn coding. If remote learning was offered, students preferred synchronous lectures over asynchronous sessions.

The number of studies to explore how challenges in an introductory programming course in the Covid era can be addressed is still limited. This work aims to provide insights of how some of these challenges in an introductory programming course could be addressed in blended learning, and to gain a better understand towards students' learning experience in different learning settings.

3 Context

3.1 The course

York College is a senior college in the City University of New York (CUNY) system, the nation's largest urban public university. York College is located in one of the most ethnically diverse counties in the country, with a student population consisting of 36.3% Black or African American, 26.3% Hispanic/Latino, 23.3% Asian, and 6.5%

White (York College, 2020). York College offers degrees in over sixty undergraduate majors and five graduate programs. Computer Science (CS) major is one of the largest majors in terms of student enrollment at York College.

CS major students at York College begin their major with an introductory programming course (CS172 – Computer Science I) that introduces them to fundamental concepts of computer programming and computational thinking. This is the first course in the programming sequence courses which students would typically take in their first year. Students are expected to complete several laboratory exercises using a high-level language such as C++. Upon completion of the course, students will be able to:

1) Understand the core concepts of structural programming and their implementation in the C++programming language;
2) Understand and demonstrate an understanding of basic C++data structures such as integers, floats, doubles, characters, Boolean, strings, and arrays;
3) Be able to code C++programs using fundamental language constructs such as conditional statements, loops, and programmer-defined functions;
4) Be able to implement basic algorithms to solve problems in C++from pseudocode or flow charts;
5) Be familiar with the run-time and development environments for C++.

Course topics include: Variables and Data Types; Arithmetic Expressions and Math functions; Inputs and Outputs; Conditionals; Iterations/Loops; User-defined functions; Recursion; Arrays; STL array and vector classes; Strings; File I/O; and Structures.

Prior to the pandemic, CS courses at York College were taught face-to-face on campus. Since Fall 2020, many courses at York college have been temporarily shifted to an online and hybrid mode (some combination of face-to-face and online components), including the mentioned introductory programming course that the author was responsible for teaching from Spring 2021 to Spring 2022 (three semesters). Due to safety concerns, the course was offered fully online in Spring 2021. All the lectures and materials took place online. In Fall 2021, the course content and practical coding exercises were online, but formative assessment took place in person. A few special tutorial sessions covering more difficult concepts were also offered in the classroom. As we are emerging from the pandemic and many classes have switched back to in-person learning, I decided to apply a blended learning approach to the section of the introductory programming course that I was responsible for in Spring 2022. The class met two times per week. The teaching and learning were evenly split between in-person and online meetings. For example, the class met in person on Tuesdays and live synchronous online on Thursdays. Each class session is 1 h and 50 min.

3.2 Tools and technologies

The main Learning Management System (LMS) used in this course is Blackboard (Blackboard, n.d.). All class materials, including course syllabus, schedules, and homework assignments, are available on the course Blackboard. Students have access

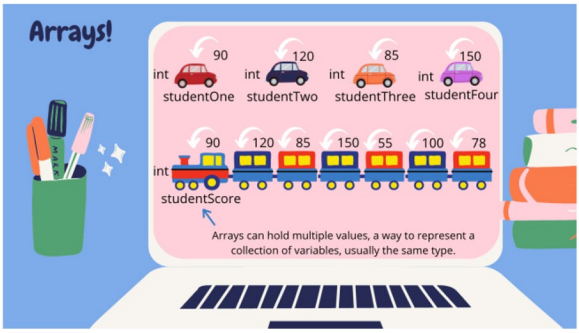

Fig. 1 Visual representation of the array concept

to online resources and recorded online lectures on the LMS. For in-person classes, we met in a computer laboratory room on campus equipped with desktop computers. Students would complete several coding exercises in a computer lab where instructors could provide instant feedback to each student. Individualized attention was possible and easy as the instructor would walk around the lab for any troubleshooting problems that students may have. In an online environment, students joined the class synchronously via an online platform (Blackboard Collaborate Ultra). Students were required to have their own computer devices and access to the Internet.

The run-time and development environment used in the course was a collaborative online Integrated Development Environment (IDE) called Replit (Replit, n.d.). An online IDE was chosen to ease the trouble of requiring students to install a version of an IDE on their systems at this stage and circumstances. Replit has a collaborative feature that allows users to share and edit the code simultaneously. This is the feature that was most often used in the class when a student would like feedback from the instructor. Since our class has over forty mini lab exercises, students were required to use Replit each every class session, whether it was offered online or in person.

The knowledge and related strategies for each programming concept were presented to students using a variety of ways. Powerpoint slides were used to introduce students to the syntax and semantics of the language (or the knowledge). The knowledge would be reinforced with visual forms. For example, Fig. 1 shows an image used to differentiate between a variable and an array.

Figure 2 was used to visualize the direction of the information flow with the use of the stream insertion operator (<<) and the stream extraction operator (>>). The explanation through visualization helps students to understand which operator to use for input or output, which is a common mistake novice learners typically make.

The course was designed to explicitly teach students to learn to code to solve programming problems, not just the syntax and semantics of the language. Several mini lab exercises, group projects, and homework assignments were used to teach students related programming strategies (i.e., know how to use and apply the concept/knowledge learned to solve a problem). For each topic, I would demonstrate the knowledge, and related strategies by coding live on an IDE. This approach helps to reinforce the new concepts learned, as well as to put them in a meaningful context.

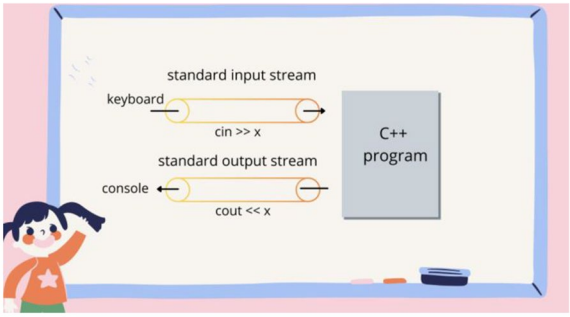

Fig. 2 Visual representation of a standard input/out stream concept

4 Reflections on pedagogical approaches in blended learning environment

Different evidence-based pedagogical approaches that were used in online and in-person classes are discussed below.

4.1 Checking for student understanding

Checking for student understanding is an important pedagogical approach, especially when a new concept is explained. In an in-person class, the instructor can make use of visual clues to see if students are struggling to understand, for example, to see students' facial expressions and their responses when asked 'Do you understand?' or 'Do you have any questions?'. A show of hands typically provides quick feedback on how the class as a whole is doing with the new materials. More formally, students can be presented with some choices and then asked to raise their hands if they agree with any of the choices. Clicker technology can also be used in the class for quick feedback.

In online classes, this becomes more challenging. Since most of the students' cameras were turned off, checking for any visual clues was impossible and impractical. Similar to a show of hands in an in-person class, a quick show of thumbs-up/thumbs-down or happy/confused emoji and a quick online poll were utilized to get a general sense of class understanding of new content. During the class session, a few low-stake class 'quiz' activities were administered each session. This is in the form of around 3–5 short multiple-choice questions. The purpose of this activity was to identify precisely what students understand and do not understand after a topic was discussed. The instructor can use the results to determine if any topics need to be retaught or revisited in the next class. This type of question is easily created using a quiz tool in the LMS, and instant feedback can be provided to students after completing the activity. Low stake 'quiz' activities were conducted about 3–4 times on average for each session. Figure 3 shows a few sample questions used for this activity.

Students were graded based on their participation and not on the results of their answers. This way, they felt safe to answer the questions in a timely fashion, and the instructor could get a real sense of how the class was doing. The use of low-stake quiz activities provides a useful insight to the instructor on the students learning and

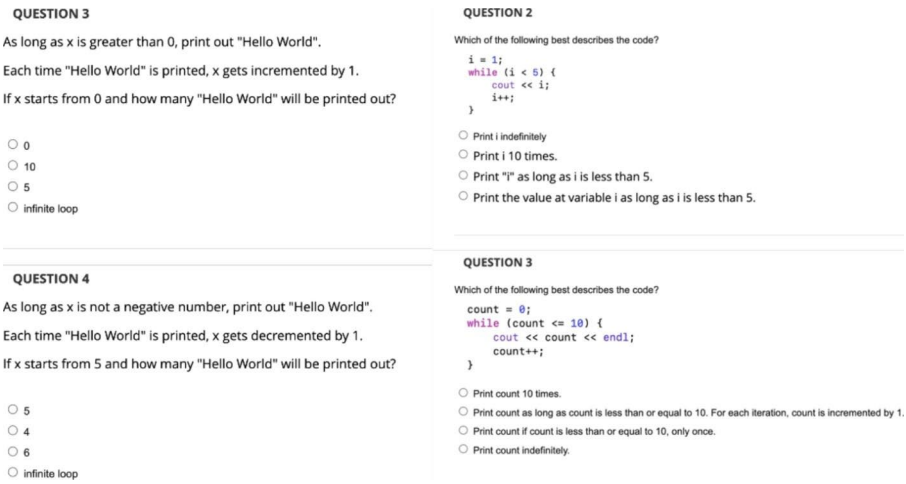

Fig. 3 Low-stake quiz activities to check for student understanding

understanding of the materials. The activities were applied to both online and in-person classes. The times required to conduct this activity in both learning modalities are very similar.

4.2 Providing meaningful feedback

Coding skills and problem-solving skills are best acquired when students have a lot of opportunities to code. This is especially important for students to learn CS1 concepts that are identified as particularly problematic, like conditionals and loops (Cherenkova et al., 2014). As suggested in Malik et al. (2022), the problem-solving skills of novice learners can be enhanced by providing them with questions that are related to the course topics. In the course, students are required to complete several lab exercises in each session. For example, during a lecture on the topic of iterations and loops, there were 4 lab exercises involved: (1) use of a for loop to list out a number sequence, (2) use of a while loop to continue to prompt input from a user until a specific character is entered, (3) use of a for loop to print out numbers in an increment of a number, and (4) use of a nested loop to print a star pattern. Typically, a set of exercises on a certain topic is completed in class within one to two sessions.

During in-person classes, students worked on the lab exercises in the computer lab. With around 25 students in the class, it was relatively easy to see who was struggling and who was able to complete the exercises. Students often asked for help to troubleshoot their code for errors. For example, in the course where C++was used, missing closing brace, missing semi-colon, missing declaration of variables, and typos of variable names are common and can be quickly spotted by the instructor when walking around the classroom. By checking and observing students work rows by rows, the instructor could quickly identify the common problems that most of the class was having. More explanation and examples would then be provided.

Online classes inevitably pose new challenges when it comes to providing student feedback. Students work on their own devices in a virtual environment. If students

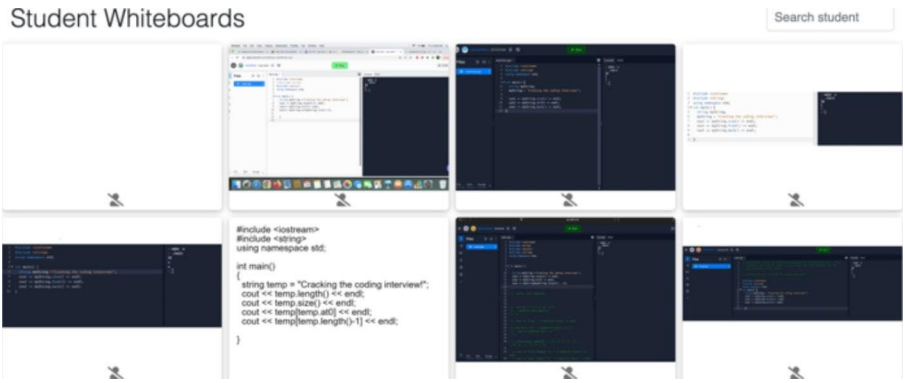

Fig. 4 Student code submissions on whiteboard.fi

were not willing to share their code with the instructor, it was impossible for the instructor to view and help troubleshoot for any errors. In addition, with the class size of 25 students, it is also impractical to screen share or view all the work individually (through a shared link or remote desktop). Since lab exercises are typically small in scale, involving around 10–15 lines of code, students were asked to copy and paste their code on an online whiteboard platform such as whiteboard.fi (Kahoot Group, n.d.) and whiteboard.chat (Whiteboard.chat., n.d.). This way, the instructor was able to view students' work all at the same time. Figure 4 shows a screenshot of a class exercise submitted by a group of students on whiteboard.fi.

The instructor could see an immediate overview of all student work in real-time. Live feedback could be provided to each student if necessary. Such online whiteboard platforms also have a feature to hide student names. Thus, I could show the work done by all students while still respecting their anonymity. The work submitted can also be a good source of discussion for students to learn various coding techniques and mistakes. I would also use this opportunity to discuss best coding practices, such as consistent indentation, commenting and documenting, and a consistent naming scheme. Furthermore, when common errors or mistakes were identified, I could address the problems right away. I find the use of an online whiteboard platform very useful for this purpose, and this is one advantage over the traditional classroom format.

Nevertheless, asking all students to share their code through an online whiteboard can be a bit of a challenge. Most students were only willing to share the code if they could complete the task, while those struggling were reluctant to share. Thus, it was harder to help students who were behind and struggling than in the face-to-face environment.

4.3 Collaborative learning

Pair programming and group work enhance the student experience in collaborative learning and improve their ability to learn how to code (McChesney, 2016; Cabrera et al., 2017). Findings by McDowell et al. (2002) concluded that students working collaboratively produced better programs and that collaborative learning is an effective

Table 2 Breakdown of group projects

Tasks	Requirements	Skills assessed
1. Input/Output	Output a welcome message. Prompt the user for the number of players. Prompt the user to guess a letter ten times. Validate user inputs.	Make use of prompts for different inputs. Determine when to use a while loop and for loop. Use conditionals to validate the user input (e.g., integer, character).
2. Gameplay structure	Create a function for input validation. Specify a player number by turn. Use a do while loop to prompt the user if they want to play again.	Organize code into functions. Understand when to use a do-while loop. Apply knowledge of math to solve a specific problem.
3. Complete Hangman Program	Use a vector to store a word list for the user to guess. Randomly choose a word from the word list. Determine if the letter guessed by the user can be found in the word. Use a vector to display the position when a letter is guessed correctly.	Apply data structures like a vector for use in different cases. Use loop with vector/array. Use String class functions. Apply a linear search algorithm to search a letter in the word.

pedagogical approach to be used in introductory programming courses. The course was designed to let students have ample opportunity to work together in small groups or in pairs to solve programming problems. Midway through the course, students were assigned to work on a course project in a group of 4–5 students. The project aimed to exercise students' coding skills with a fun and motivating task, in this case, to create a Hangman program. The Hangman program lets students use and apply several knowledge and concepts learned to solve a meaningful and familiar problem. For example, students need to apply loops to keep prompting a user to try to guess one letter at a time. Students use conditional statements to check if the guessed letter occurs in the chosen word. Students need to also apply the knowledge of array and String classes to draw dashes and fill in the blanks with the guessed letter in the right positions. The project is broken down into three small and manageable tasks, as shown in Table 2.

During in-person classes, students would have the opportunity to discuss and work together during class hours. The face-to-face interactions allowed team members to share different approaches to complete the required tasks. Typically, one person would work on the code while others review and make suggestions. I would visit each group and see their progress. Tips or hints would be provided for the group to be able to proceed on their own as a group.

In online classes, group discussion is facilitated by using a variety of online tools, such as group breakout sessions, discussion forums, and shared digital documents.

Similar to in-person classes, when working on a coding problem, one group member would share his/her screen during a group breakout session. Everyone was encouraged to contribute to the work. In an online environment, I would join each group breakout session to provide them with feedback. However, it was slightly more time-consuming to provide the same types of feedback to each group.

5 Student perceptions Survey

5.1 Method

An online survey was administered towards the end of the introductory programming course in Spring 2022 to all students enrolled. The purpose of the survey is used to help us understand student perceptions towards remote learning and in-person modality used within a section of the introductory computer programming course at York College. The survey contains six multiple choices, thirteen five-point Likert scale questions, and three open-ended questions.

5.2 Participations and data collection

There were 29 students at the beginning of the course, two withdrew, and two stopped attending the course or never attended the course, with a total of 25 active enrolled students remaining. All actively enrolled students agreed to participate in the study on a voluntary basis. An online survey was distributed to them via the LMS by the course instructor. Participation in the survey was completely anonymous. Everyone responded to the survey, resulting in a response rate of 100%.

6 Results

As shown in Tables 3 and 68% of respondents are CS major students, 8% are from Information System Management major, 16% from other STEM majors, and the rest (8%) from non-STEM majors. The respondents were largely male (84%). The majority of the class (84%) stated that this course was their first computer science programming course at a college level. Two students that said they had taken a programming course prior either withdrew or failed the course in the previous semester.

Table 4 shows the results of students' perceptions of activities and learning experience in an in-person environment from the five-point Likert scale questions, whereas Table 5 shows the results of the same five-point Likert scale questions in an online environment. The questions focused on students' perceptions of their engagement, learning behavior, satisfaction, and expectations of the two different learning modalities.

From Table 4, more than half of the students (60%) strongly agreed with the statement 'I find in-person lab exercises effective in learning programming skills.' About half of the students (48%) strongly agreed with most of the statements on in-person learning and experience. The lowest percentage (28%) of all the items that students

Table 3 Participant Demographics

	Total Sample
Major (N=25)	
Computer Science	68%
Information System Management	8%
Other STEM Major	16%
Other non-STEM Major	8%
Gender (N=25)	
Male	84%
Female	16%
Preferred not to say	0%
Race/Ethnicity (N=25)	
American Indian or Alaska Native	0%
Asian	40%
Black or African American	28%
Hispanic/Latino	12%
Native Hawaiian or Other Pacific Islander	4%
White	16%
First-time programming course at the college level (N=25)	
Yes	84%
No	16%

strongly agreed on is the statement 'I completed lab exercises within the class hour in the vast majority of in-person classes.' From Table 5 for online learning and experience, about half (48%), which is the highest percentage of all the items that students strongly agreed on, is to the statement 'I attended the vast majority of online classes.' The lowest percentage (20%) is the same as the in-person environment, which is 'I completed lab exercises within the class hour in the vast majority of online classes.' The percentage of items that students strongly agreed to in an online environment is slightly lower than in an in-person environment across all the items.

Table 6 provides another view of the same set of Likert scale questions presented in Tables 4 and 5.

As we can see from the results, a higher number of students agreed or strongly agreed with 'in-person classes' modality than 'online classes' in almost all aspects. The two largest gaps identified were item number 9 on getting feedback in the class and item number 2 on class participation. 96% of respondents agreed or strongly agreed that the feedback that they get in an in-person class helped clarify things that they did not understand, while 64% agreed or strongly agreed that the feedback that they get in an online class helped clarify things that they did not understand. 84% of respondents agreed or strongly agreed that they actively participated in the vast majority of in-person classes, while only about half (56%) said so for online classes. The items that are slightly similar in terms of what students agreed or strongly agreed on in the two learning environments were item 1 on attendance and item 4 on satisfaction with the coursework. It is also worth noting that the number of students who agreed or strongly agreed on the overall expectations of the course in the vast majority of in-person and online classes correspond to the learning outcomes and subject content is very much similar, with 80% for in-person and 72% for online classes.

Table 4 Perceptions of students towards in-person activities and learning experiences (N = 25)

Modality	SA	A	N	D	SD
1) I attended the vast majority of in-person classes.	56%	20%	12%	8%	4%
2) I actively participated in the vast majority of in-person classes.	48%	36%	12%	4%	0%
3) I was fully engaged in the vast majority of in-person classes.	48%	40%	8%	4%	0%
4) I was satisfied with my in-person coursework.	48%	32%	16%	4%	0%
5) I find in-person lab exercises effective in learning programming skills.	60%	36%	4%	0%	0%
6) The lectures conducted in in-person classes were engaging	48%	40%	12%	0%	0%
7) I completed in-class activities within the class hour in the vast majority of in-person classes.	36%	56%	8%	0%	0%
8) I completed lab exercises within the class hour in the vast majority of in-person classes.	28%	56%	12%	4%	0%
9) The feedback that I get in an in-person class help clarify things that I did not understand.	48%	48%	4%	0%	0%
10) I find the vast majority of in-person classes intellectually stimulating.	48%	32%	20%	0%	0%
11) Overall, my expectations of the course in the vast majority of in-person classes correspond to the learning outcomes and subject content.	48%	32%	20%	0%	0%

SA: Strongly Agree. A: Agree. N: Neutral. D: Disagree. SD: Strongly Disagree

Table 7 shows the overall satisfaction and experience in in-person and online classes. The overall satisfaction of students with the in-person and online classes are very similar. 84% of respondents were satisfied or very satisfied with in-person classes, while 80% were satisfied or very satisfied with online classes. For overall experience, over half of the respondents (52%) said that the experience of in-person classes was great, while 40% stated that the experience of online classes was great. 2 respondents (8%) stated that the in-person experience was bad.

Table 8 shows the results of students when asked if there is any difference in how they were learning the materials in in-person classes or in online classes. 68% answered ‘yes, there were some differences’, and 32% said ‘no’.

A follow-up question was asked on why they think that there is a difference. Among those who said yes, those who preferred in-person classes mentioned that (1) in-person classes allow them to ask the instructor more questions and get better feedback, (2) in-person classes can keep them more engaged and focused, and (3) in-person classes facilitate interactions with their peers. For example, some students said:

I am able to fully understand and complete the work better in person than online.

Table 5 Perceptions of students towards online activities and learning experiences (N = 25)

Modality	SA	A	N	D	SD
1) I attended the vast majority of online classes.	48%	32%	16%	0%	4%
2) I actively participated in the vast majority of online classes.	40%	16%	36%	8%	0%
3) I was fully engaged in the vast majority of online classes.	28%	48%	24%	0%	0%
4) I was satisfied with my online coursework.	40%	44%	16%	0%	0%
5) I find online lab exercises effective in learning programming skills.	44%	36%	16%	4%	0%
6) The lectures conducted in online classes were engaging.	36%	32%	32%	0%	0%
7) I completed in-class activities within the class hour in the vast majority of online classes.	36%	36%	20%	8%	0%
8) I completed lab exercises within the class hour in the vast majority of online classes.	20%	52%	24%	4%	0%
9) The feedback that I get in an online class help clarify things that I did not understand.	24%	40%	28%	8%	0%
10) I find the vast majority of online classes intellectually stimulating.	24%	36%	40%	0%	0%
11) Overall, my expectations of the course in the vast majority of online classes correspond to the learning outcomes and subject content.	40%	32%	28%	0%	0%

SA: Strongly Agree. A: Agree. N: Neutral. D: Disagree. SD: Strongly Disagree

> In in-person class you get to interact with the professor better, the professor walks around and checks your work and all.
> In person was more engaging as I was able to ask questions and receive immediate help when I need assistance in my coding.
> I like the in-person better because it makes me more active, and I'm able to participate by asking questions.
> I was able to be more focused in person than when I am at home, and everything seemed clearer.
> The main differences to me is that it's easier to ask questions when we're able to see each other in person, and the classes in person overall move along better.

While those who preferred online classes expressed that (1) online classes allow them to go back and review the class recordings, and (2) online classes allow them to work at their own pace. For example:

> With online, you get the recording so at least you can go back and check what you miss heard or get a clearer understanding.

Table 6 Comparison of students' perceptions in in-person and online learning environments (N = 25)

Modality	Strongly Agree/ Agree	Neutral	Strongly Disagree/ Disagree
1) I attended the vast majority of [in-person/online] classes.			
In-person	76%	12%	12%
Online	80%	16%	4%
2) I actively participated in the vast majority of [in-person/online] classes.			
In-person	84%	12%	4%
Online	56%	36%	8%
3) I was fully engaged in the vast majority of [in-person/online] classes.			
In-person	88%	8%	4%
Online	76%	24%	0%
4) I was satisfied with my [in-person/online] coursework.			
In-person	80%	16%	4%
Online	84%	16%	0%
5) I find [in-person/online] lab exercises effective in learning programming skills.			
In-person	96%	4%	0%
Online	80%	16%	4%
6) The lectures conducted in [in-person/online] classes were engaging.			
In-person	88%	12%	0%
Online	68%	32%	0%
7) I completed in-class activities within the class hour in the vast majority of [in-person/online] classes.			
In-person	92%	8%	0%
Online	72%	20%	8%
8) I completed lab exercises within the class hour in the vast majority of [in-person/online] classes.			
In-person	84%	12%	4%
Online	72%	24%	4%
9) The feedback that I get in an [in-person/online] class help clarify things that I did not understand.			
In-person	96%	4%	0%
Online	64%	28%	8%
10) I find the vast majority of [in-person/online] classes intellectually stimulating.			
In-person	80%	20%	0%
Online	60%	40%	0%
11) Overall, my expectations of the course in the vast majority of [in-person/online] classes correspond to the learning outcomes and subject content.			
In-person	80%	20%	0%
Online	72%	28%	0%

I was able to refer to any info that was given in the online meeting because it was recorded, so if I had any questions, I can just refer to the point based on what was the topic being discussed at a certain point.

Table 7 Overall satisfaction and experience in in-person and online classes (N = 25)

Modality	Very Satisfied	Satisfied	Neutral	Unsatisfied	Very Unsatisfied
	Great	**Good**	**Okay**	**Bad**	**Terrible**
Overall, how satisfied or unsatisfied were you with the [in-person/online] classes.					
In-person	56%	28%	12%	4%	0%
Online	44%	36%	16%	4%	0%
Overall, how would you rate your experience of [in-person/online] class.					
In-person	52%	28%	12%	8%	0%
Online	40%	32%	28%	0%	0%

Table 8 Opinions on learning differences between the two modalities

	Total Sample
Is there any difference in how you were learning the materials in in-person classes and in online classes?	
Yes, there are some differences in how I was learning the materials.	68%
No, there is no difference in how I was learning the materials.	32%
Is there any difference in how you were obtaining coding skills in in-person classes and in online classes?	
Yes, there are some differences in how I was obtaining coding skills.	32%
No, there is no difference in how I was obtaining coding skills.	68%

> Online learning allows me to pace myself more appropriately to my needs.
> Online classes get recorded, so you can always go back to watch them, but this is not true for in-person.

Table 8 also shows the results of students when asked if there is any difference in how they were obtaining coding skills in in-person classes and in online classes. 32% answered 'yes, there were some differences' while 68% said 'no'. A follow-up question was asked on why they think that this is a difference. Those that said yes all think that in-person was better, and here are a few examples:

> Talking to classmates in person makes it easier to understand and comprehend things.
> I found more people to work together in the group activities.
> As I progressed in my coding I found it helpful that I could ask my peers or professor for assistance in my code, I also found it easier to concentrate during coding since I am surrounded by a learning environment.
> I seem to code better in my in-person classes compared to my online class
> Some examples of specific topics were easier in a specific setting but not both online or in person.

Another open-ended question asked students about their group experience. Here are some of the responses:

> I found more interest in this subject and I have learned from how work as a team.

I learned a lot from my fellow group mates especially when we met on Blackboard to collaborate and explain the code to each other.
I've learned that it is always good to ask others for help when you need it.
I leaned how to break the task down to come up with a solution and work on it step by step.
I learned that each people's way of coding are not the same. It was helpful for me to get to kow how to code in different ways.
I prefer doing group projects in person, so then we can see who needs help in which area.

Students' comments in response to open-ended questions provide useful insights into the difference in the learning modality preferences. A few students mentioned that in an online environment, the recordings were made available, and those were useful when they wanted to review the materials taught at their own pace. Those who preferred in-person classes mentioned that student-instructor and peer interactions were better. They were able to learn more from their peer, and it was easier to get feedback from the instructor. They were also more focused and engaged in an in-person environment.

7 Discussions and conclusion

The pandemic has brought changes in higher education. Blended learning or hybrid learning has become more popular and more practical because it allows instructors greater flexibility in how they choose to implement instructional approaches, especially when health and safety are still a concern. The goal of this study is to understand students' points of view and preferences in the two different learning environments in an introductory programming course in which several evidence-based pedagogical approaches, for both online and face-to-face, have been used in the class to promote learning for novice programmers.

In teaching introductory programming courses, checking for student understanding is an essential pedagogical approach, especially when a new concept is explained. In addition to online polling and asking class questions, low-stake class activities, in the form of multiple-choice questions, were used in every class, whether in-person or online. The quiz tool built into the LMS allows us to ask questions that can be automatically graded. The quick assessment of students' understanding enables us to understand which new concepts students struggle with. Any concepts that students have yet to grasp will be explained or revisited with more examples. Students in an introductory programming course should also have ample opportunity to practice coding skills. Learning to program involves complex and variety of cognitive activities (Robins et al., 2003). Novice programmers not only need to learn the new concepts, but they should be able to use and apply the concepts to solve problems. Assignments and class exercises should be purposefully designed to incorporate critical thinking skills, problem-solving skills, and other programming-related skills to the problems. This would enable novice learners to focus on the problem-solving aspects of the program and learn to apply the new language constructs, such as loop,

and conditional statement, in meaningful contexts. Nevertheless, the logic or semantic errors in a program can be hard and frustrating for a novice learning to identify and fix (Ettles et al., 2018). In my own experience, students are also struggling to understand and fix syntax errors generated by the compiler. This requires practice and experience for them to identify and fix the problems on their own. Therefore, with proper guidance and feedback from the instructor, the learning process of students can be less painstaking. Meaningful and timely feedback from the instructors would alleviate students' frustration in the process. Providing feedback in a face-to-face setting is relatively straightforward. Nonetheless, for an online environment, this can be done by meticulously incorporating instructional approaches that facilitate feedback within the synchronous learning environment. Furthermore, collaborative learning such as pair programming and group project helps boost students' ability to learn to code. Students can learn from their peers. Groups are carefully assigned halfway through the course to incorporate diversity by recognizing each group member's different backgrounds and skills. Nowadays, the LMS has features that allow group members to discuss, share ideas, and work collaboratively as a team.

Survey data provide useful insight into students' perceptions of how they learn topics and skills taught in an introductory programming course in both modalities. The satisfaction and expectation levels from the course are very similar in both environments. The data from the survey confirmed that the face-to-face interactions, immediate feedback, and the ability to engage better were what most students find advantageous in in-person classes in learning and practicing coding. Online components offer students greater flexibility and convenience. Open-ended survey questions provided additional insight into why some students favored in-person classes and why some still prefer online learning. From the results of the survey, while most preferred in-person activities more, a few strongly preferred online learning over in-person classes. Although the classes are designed to meet synchronously online, students still think that the online environment provides them with greater flexibility and convenience. They could review the materials at their own pace and time; not having to commute was an additional benefit.

As for the author's experience for the past two years, troubleshooting student work and helping with coding logic in online classes were the biggest challenges in an introductory programming course, especially when the course is the first coding class for almost all the students. The lack of direct in-person assistance or supervision in online classes is a concern (Hamid and Rashid, 2020). Students may also have difficulty staying focused and motivated and feel isolated in an environment where they do not have to be physically present in the classroom (Hamid and Rashid, 2020; Tian 2021; Ismail & Razak, 2021). As Lohiniva and Isomöttönen (2021) stated, meeting the instructor face-to-face is important because it makes students feel more connected in learning which could enhance their retention. Having said that, I firmly believe that the online components integrated into the class learning environment were very beneficial during the COVID-19 pandemic and beyond. Online resources, including the course syllabus, learning outcomes for each topic, lecture materials, over 40 exercises, and sample solutions, were made available online within the course LMS. Students can easily access them and review the materials at their own pace and time outside class hours. Moreover, each student is unique and has a different

learning style. Those who prefer to learn in an online environment find it easier to navigate the course contents and complete course exercises. It is also worth noting that the class attendance in online classes, on average, was about 25% higher than in in-person classes. Students' reasons for absences that were common were illness and quarantines. When classes are conducted online, students can still join the sessions even when they are not well or exposed to COVID-19. When comparing students' performance with grades obtained from the course across the three semesters (Spring 2021-Spring 2022), there was an overall increase in students' grades from Fall 2021 (fully online with in-person assessments) to Spring 2022 (blended learning). For example, there was a 44% increase for students who received an 'A' grade range. However, at this point, it is still difficult to draw a conclusion if different modalities could improve students' assessed learning outcomes as the student populations are different.

In this paper, I have shared my experience in introducing teaching interventions with the goal of trying to improve students learning and experience in an introductory programming course at York College, CUNY. Through a careful design and implementation of pedagogical approaches used in the class, novice learners could potentially benefit from both face-to-face and online components of blended learning. As stated in Alammary (2019), a balanced and meaningful mix of the different components in introductory programming courses can bring advantages from both online and face-to-face components. The findings from the survey are well-aligned with previous studies. However, because of the small sample size, I was not able to present the results with statistical significance. Future work will further investigate the impacts on students learning outcomes in different learning environments with a larger sample size and not just based on their perceptions.

Funding No funding was received to assist with the preparation of this manuscript.

Availability of data and material Not applicable.

Available of data and material The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflicts of interest/Competing interests The authors declare that they have no competing interests.

References

Alammary, A., Carbone, A., & Sheard, J. (2017). Curriculum transformation using a blended learning design toolkit. In *40th HERDSA Annual International Conference*

Alammary, A. (2019). Blended learning models for introductory programming courses: A systematic review. *PloS one*, *14*(9), e0221765

Blackboard. (n.d.) Blackboard website. Retrieved May 8 (2022). from https://www.blackboard.com/

Cabrera, I., Villalon, J., & Chavez, J. (2017). Blending communities and team-based learning in a programming course. *IEEE Transactions on Education*, *60*(4), 288–295

Capdevila, L. R. (2021, July). Challenges and Successes of the Transition to Online Format of a Lower Division Aerospace Engineering Class during COVID-19. In *2021 ASEE Virtual Annual Conference Content Access.*

Cherenkova, Y., Zingaro, D., & Petersen, A. (2014, March). Identifying challenging CS1 concepts in a large problem dataset. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 695–700)

CUNY. (2020). CUNY's 275,000 Students And 20,000 Faculty Resume Spring Semester Via Distance Learning. Retrieved May 2, 2022, from https://www1.cuny.edu/mu/forum/2020/03/19/cunys-275000-students-and-20000-faculty-resume-spring-semester-via-distance-learning/

Ettles, A., Luxton-Reilly, A., & Denny, P. (2018, January). Common logic errors made by novice programmers. In *Proceedings of the 20th Australasian Computing Education Conference* (pp. 83–89)

Graham, C. R. (2006). Blended learning systems. *The handbook of blended learning: Global perspectives, local designs, 1*, 3–21

Hamid, F., & Rashid, F. (2020, October). Adjusting to the new normal: perspectives from an introductory programming sequence course. In *Proceedings of the 9th Computer Science Education Research Conference* (pp. 1–2)

Ismail, N. Z., & Razak, M. R. (2021). The Challenges of Learning Programming Subject in Online Distance Learning (ODL) Environment at UiTM Pahang. *Gading Journal of Science and Technology (e-ISSN: 2637-0018)*, *4*(2), 27–31

Kahoot Group. (n.d.). Whiteboard.fi - Free online whiteboard for teachers and classrooms. Retrieved March 8 (2022). from https://whiteboard.fi/

Kaur, M. (2013). Blended learning-its challenges and future. *Procedia-social and behavioral sciences*, *93*, 612–617

Lee, N. T. S., Kurniawan, O., & Choo, K. T. W. (2021, June). Assessing Programming Skills and Knowledge During the COVID-19 Pandemic: An Experience Report. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1* (pp. 352–358)

Lohiniva, M., & Isomöttönen, V. (2021, October). Novice Programming Students' Reflections on Study Motivation during COVID-19 Pandemic. In *2021 IEEE Frontiers in Education Conference (FIE)* (pp. 1–9). IEEE

Luxton-Reilly, A., Simon, Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., Ott, L., Paterson, J., Scott, M., Sheard, J., & Szabo, C. (2018, July). Introductory programming: a systematic literature review. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (pp. 55–106)

Malik, S. I., Mathew, R., Al-Sideiri, A., Jabbar, J., Al-Nuaimi, R., & Tawafak, R. M. (2022). Enhancing problem-solving skills of novice programmers in an introductory programming course. *Computer Applications in Engineering Education*, *30*(1), 174–194

McChesney, I. (2016, February). Three years of student pair programming: action research insights and outcomes. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 84–89)

McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2002, February). The effects of pair-programming on performance in an introductory programming course. In *Proceedings of the 33rd SIGCSE technical symposium on Computer science education* (pp. 38–42)

Medeiros, R. P., Ramalho, G. L., & Falcão, T. P. (2018). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, *62*(2), 77–90

Mehmood, E., Abid, A., Farooq, M. S., & Nawaz, N. A. (2020). Curriculum, teaching and learning, and assessments for introductory programming course. *Ieee Access : Practical Innovations, Open Solutions*, *8*, 125961–125981

Omer, U., Farooq, M. S., & Abid, A. (2021). Introductory programming course: review and future implications. *PeerJ Computer Science*, *7*, e647

Rahman, M. M., Paudel, R., & Sharker, M. H. (2019, October). Effects of infusing interactive and collaborative learning to teach an introductory programming course. In *2019 IEEE Frontiers in Education Conference (FIE)* (pp. 1–8). IEEE

Replit. (n.d.). Replit website. Retrieved May 8 (2022). from https://replit.com/~

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, *13*(2), 137–172

Seeling, P. (2020, October). Switching to stay home instruction: Impacts of the coronavirus pandemic on learner performance for an introductory computer science course. In *Proceedings of the 21st Annual Conference on Information Technology Education* (pp. 294–294)

Tayebinik, M., & Puteh, M. (2013). Blended Learning or E-Learning? *International Magazine on Advances in Computer Science and Telecommunications*, *3*(1), 103–110

Tian, T. (2021). Hybrid Teaching of Introductory Programming during the COVID-19 Pandemic–a Case Study. *International Journal of Academic Research in Education and Review*, *9*(7), 297–303

Toti, G., & Alipour, M. A. (2021). Computer science students' perceptions of emergency remote teaching: An experience report. *SN Computer Science*, *2*(5), 1–9

Vollbrecht, P. J., Porter-Stransky, K. A., & Lackey-Cornelison, W. L. (2020). Lessons learned while creating an effective emergency remote learning environment for students during the COVID-19 pandemic. *Advances in physiology education*, *44*(4), 722–725

Whiteboard.chat. (n.d.). Whiteboard.chat - Interactive Online Whiteboard for Teachers and Students. Retrieved May 26 (2022). from https://www.whiteboard.chat/

Xinogalos, S. (2016). Designing and deploying programming courses: Strategies, tools, difficulties and pedagogy. *Education and Information Technologies*, *21*(3), 559–588

Yamamoto, M., Sekiya, T., & Yamaguchi, K. (2011, August). Relationship between programming concepts underlying programming skills. In *2011 International Conference on Information Technology Based Higher Education and Training* (pp. 1–7). IEEE

YeckehZaare, I., Grot, G., Dimovski, I., Pollock, K., & Fox, E. (2022, February). Another Victim of COVID-19: Computer Science Education. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1* (pp. 913–919)

York College. (2020). *Annual Factbook Fall 2020*. York College, CUNY

Zota, R. R., & Granovskiy, B. (2021). Remote Learning for K-12 Schools during the COVID-19 Pandemic. CRS Report R46883, Version 3. *Congressional Research Service*. https://sgp.fas.org/crs/misc/R46883.pdf