Online Computing Summer Camp with Non-Verbal Students with Autism Spectrum Disorder

Tuba Ketenci  
H. Milton Stewart School of Industrial & Systems Engineering  
Georgia Institute of Technology  
Atlanta, GA, USA  
tuba.ketenci@isye.gatech.edu

Claudia Dwertz  
H. Milton Stewart School of Industrial & Systems Engineering  
Georgia Institute of Technology  
Atlanta, GA, USA  
cdwertz@gatech.edu

Nurjamal Chonoeva  
National College of Education  
Chicago, IL, USA  
nchonoeva@my.nl.edu

DOI: 10.13140/RG.2.2.34119.16803  
Keywords: Autism, Coding

In recent years, substantial growth in the STEM fields coincides with calls for more STEM workers (Holdren, Lander, & Varmus 2010). However, despite this need, the demand for STEM employees in the United States remains unmet (Taylor 2016; U.S. Congress Joint Economic Committee 2012; Xue & Larson, 2015). Further complicating the issue, the participation of women, minorities, and individuals with disabilities (IWD) in STEM fields remains discouragingly and persistently low (National Science Foundation [NSF] 2017). The recent NSF report, “Women, Minorities, and IWDs in Science and Engineering,” shows the persisting underrepresentation of several groups in STEM fields, particularly IWDs, who constituted only 7% of the workers in science and engineering (2017).

To address this unmet need and to increase diversity in STEM fields, scholars have explored various STEM-related initiatives. A particularly noteworthy one relates to computing education in formal and informal learning environments (e.g., Mouza, Marzocchi, Pan, & Pollock, 2016; Starrett, Doman, Garrison, & Sleigh, 2015; Piech et al., 2019). Findings from a recent literature review and several other studies indicate a substantial increase in computing education studies about redressing gender- (Authors 2022; Suajani 2017; Hicks 2019) and race-based inequalities (Scott, Sheridan, and Clark, 2015; Eglash, Gilber, Taylor & Geier, 2013). On the other hand, studies focusing on IWDs remain scarce (Margeliux, Ketenci, & Decker, 2019; Israel et al. 2015; Stefik, Ladner, Allee, & Mealin, 2019).

Motivated to fill this gap, we developed an online, flipped summer computing camp with a data science focus for students with autism spectrum disorder (ASD). Our overarching aim in this study was to examine the impact of computing instruction on participants’ acquisition of computing concepts and engagement with computing activities. Given that ASD is a highly prevalent and severe neurodevelopmental disorder resulting in a substantial burden for individuals, families, and society, the demand for quality instruction for students with ASD is critical. Unfortunately, computing is one of the disciplines receiving inadequate attention in research about students with ASD. Lack of instruction in such a needed skill in the current labor market might raise additional barriers to a well-paying job, beyond their existing challenges with social interaction, communication, collaboration, repetitive behavior, and limited interest (American Psychiatric Association, 2013).

Computer Science with Students with ASD

Students with ASD are increasingly part of inclusive classroom environments. Their placement in general education settings is increased from 4.8% to 36% from 1991 to 2013 despite the fact it varies considerably across states in the United States (Barnett & Cleary, 2015).
Based on this trend, students with ASD face an expectation to learn the same content and perform on levels similar to the typical student population. However, most of the educational programs for the ASD population focus on communication, collaboration, life, and functional skills rather than academic content. Among instructional programs, the focus was verbal literacy and mathematics, not computing.

Computing is one of the fastest-growing industries. Research findings indicate that employment in computing occupations could grow 11% between 2019 and 2029, much faster than the average for all occupations (U.S. Bureau of Labor Statistics, 2021). Developing computer science (CS) skills can lead to additional opportunities for higher education and a well-paying job for individuals with ASD. Unfortunately, the employment rate among adults with ASD is abysmally low, with approximately 82% being unemployed (Taylor & Seltzer, 2011).

Computing is an area in which people with ASD could shine due to their greater aptitude to systemize than to empathize. This ability to systemize aligns with fields or interests that require analytical thinking (Baron-Cohen, 2019). In addition, they are a good fit for CS jobs because they have “desirable quantities employers look for, such as careful attention to detail, commitment to high quality and accuracy, box thinking, conscientiousness and diligence, and the ability to work independently” (Felicetti, 2020). Based on these strengths, some technology companies have started programs to hire students with ASD (e.g., Microsoft and SAP).

To prepare students with ASD for jobs that require computing skills, educators need to offer CS instruction tailored to the needs of this population. However, the number of CS educators across the nation is relatively low, and among those, very few are ready to teach students with disabilities, particularly students with ASD. Complicating the issue further, only fourteen empirical studies about computing education feature students with ASD, nine of which involve the research groups of Dr. Israel and Dr. Lindsay. The other five are the work of individual scholars.

Studies suggest that when students’ individual needs are met through adapted CS programs in K-12 teaching, their possible proclivity for computing education may increase (Lindsay & Hounsell, 2017; De-Lawrence et al., 2021). One way to promote student desire to learn computing is introducing this field at an early age (Knight, Wright, & DeFreese, 2019). Helping students with ASD to foster an interest in computing early on may increase their likelihood of succeeding in computing (Lindsay & Hounsell).

One common finding of these studies is the lack of research on computer science instruction designed for students with ASD (Knight, Wright, & DeFreese, 2019). Students with ASD require individualized support and when it is not provided students of this group may face various challenges in mastering computing skills (Israel et al., 2015). Documenting these challenges may lead to new tools and curricula to better support students with ASD (Koushik, Kane & Kane, 2019).

**Online Teaching CS**

One way to increase participation among students with disabilities in CSed is to deploy more online learning environments (OLE). These teaching platforms have many advantages, such as autonomy and flexibility in the learning process and reduction in the stigmas that IWDs tend to experience (Greer, Rice, and Dykman 2014). According to a recent literature review about online learning among K-12 students, IWDs perceive that OLEs promote their learning (Harvey, Greer, Basham, and Hu 2014; Beck, Maranto, and Lo 2014; as cited by Rice and Dykman 2018). Despite these benefits, few scholars have conducted studies focused on the
development of OLEs and the assessment of their impact on students with disabilities, especially in K-12 computing education (Greer, Rice& Dykman 2014). Given the exponential increase in technological improvements over the last decade, growth in the number and availability of structured OLEs could partially redress the underrepresentation of IWDs in CS fields.

Among the computing studies conducted with students with ASD, only one study took place in an OLE (Begel et al. 2020). In that study, the authors taught participants computing concepts through game-building activities in a visual block-based context via video call meetings. In addition to delivering the computing curriculum, they tried to improve the communication and teamwork skills of the students. Their findings indicate positive results. Another study with students with ASD provided video-based instruction to teach computing concepts. The findings indicate that a video prompting intervention helped participants grasp block-based coding and acquire all of the target skills (Wright 2019). However, the study was not conducted in an OLE.

Similar to CS education, the recent overall review of literature for online studies with a population with ASD underlined the scarcity of online learning studies with students with ASD. Only four studies were found in higher education studies (Newman et al., 2011). The total of participants of those studies was 22 and they had conflicting findings. Clearly, more research is needed to understand the impact of online learning on students with ASD’s understanding of the concepts and engagement with the course material.

Hence, the aim of the current study was to build online computing course and extend the use of block-based programming to teach computing concepts to students with ASD in an online environment in a flipped format. Moreover, the specific objectives of the study were to evaluate the effectiveness of the program among students with ASD on their grasp of the concepts and engagement with the content in a two-week-long summer camp. The following research questions guided the study:

RQ1: Does the fully online, flipped computing summer camp facilitate the acquisition and development of computing concepts?
RQ2: To what extent do participants enjoy and feel engaged in the computing instruction in the fully online, flipped summer computing camp?

2. Method

2.1 Participants

The target audience of the camp was high-school students with ASD and little or no experience in CS and computer programming.

2.1.1 Ali: From Dubai, Ali was 29 years old when he enrolled in the summer camp. As a non-verbal autistic student on the spectrum, Ali was in the “severe” classification because he needed constant aid. He received support with communication and using a computer from his mother and father. Until recently, Ali’s parents thought he had a very low IQ and could not communicate at a high language level. Once his sister introduced them to the rapid prompting method, they discovered his actual level of intelligence and capability. Ali and his parents explored CS for the first time during our study. His sister had a CS degree, contributing to his interest in CS.

2.1.2 Max: A Caucasian autistic man, Max was 25 years old when he enrolled in the summer camp. He was also taking other classes, including a supply chain class. Max was a non-verbal autistic student on the spectrum; he received support with communication. Because he could not easily use a computer mouse or trackpad, his parents were always with him during the camp, helping him use the computer. Max’s parents were CS professionals, and they helped Ali and his
parents by holding tutoring sessions during the camp. We called Max’s parents “lead parents” in this paper to distinguish them from Ali’s parents.

2.2 Study Design
2.2.1 CS Content. The curriculum included computer programming basics through Snap!, a visual block-based programming tool. The content fit into a set of modules containing videos, projects, and resources. Participants followed the sequence in each course module, listening to the lectures and completing projects. The camp curriculum featured three modules, each one more challenging. The first module, “Carol the robot,” covered the “basics” of computing: (a) what a program is and (b) some of the main control structures. The second module, “Programming in Snap!,” captured the essence of computing: (a) data structures, and (b) recursion. The third module covered the application of computing to data science. However, completion of this third level was beyond the scope of the current study.

2.2.1 Instructional Design. The curricula employed in this study were designed based on explicit instruction. According to a recent literature review, explicit instruction has five essential components (Archer & Hughes, 2010). These five components are as follows: (1) breaking down the complex task into manageable subtasks, (2) modeling the content or skill with precise descriptions and (3) demonstrations, (4) promoting engagement through gradually faded scaffolding, and (5) proving feedback and providing purposeful opportunities to students to demonstrate their learning. In terms of the explicit instruction components, the curricula applied in this study differed in the way promoting engagement gradually faded scaffolding in accordance with the fourth component. We kept highly scaffolding instruction while were increasing the challenge with the projects.

The instructional design for the camp was the flipped classroom, meaning students interact with new content in an online environment asynchronously before attending faculty-led instruction in a synchronous session. Class time permits application of the newly learned information. In our intervention, we conducted two synchronous sessions every day with the participants, one before and one after an asynchronous learning period. First, we conducted a morning meeting using a video call to set the goal for that day and to address any questions or concerns. Students then watched the instructional videos developed by the subject-matter expert and worked on the mini-assignments or lab activity. After this asynchronous learning time, we met with the students and parents again on a video call, reviewed their work, and provided feedback. During the synchronous meetings, we offered positive, verbal reinforcement to the participants.

Development of the instructional videos for the camp followed the principle of video modeling. The subject matter expert recorded videos while explaining the concept using PowerPoint slides. The instructor also kept his camera on so that students could see who was talking during the recording.

The research team prepared mini-assignments and lab activities for the students in a step-by-step fashion so that they could analyze each task assigned during the camp. All of the materials are posted on the camp’s Canvas page. In addition to videos, slides, and task analysis, we also developed code outlines to accompany each instructional video. Please see Figures 1 for the code outlines developed for the first module.
In addition to our meeting with the participants and their parents, one of the parents took the lead and did additional teaching through one video call per day. During each session, the lead parent explained and asked specific questions to lead the students to the correct answer for the assignments. For example, when a student was stuck on one part of the code, the lead parent explained the process to the student:

So you’re going to repeat going to a wall and [then] turn left until you are in a place where the beeper is. So I think if we put “repeat until” beeper here, and then that other thing repeats until the front is blocked, turn left.

2.2.3 Communication

The parents, students, and instructor constantly communicated throughout the synchronous sessions to teach computing and help with the assignments. The parents had to use the spelling board to communicate with their child, and the students often answered quickly and correctly using the board. Here is an example of a parent summarizing how the student communicated with them:

I read him the task, and I say, “what do we do first? … How do we find the beeper in the corner?” And he said to move forward until she is blocked. So I put to do this until and then put in “blocked” and put in “move forward.” Then I said, “what’s next?” And he typed, “turn left,” and then I said, “okay, what’s next?” And he typed, “move forward until blocked.” I said, “okay, what’s next?” And then I said, “what kind of command are we going to use next?” And he typed, “loop.” Then he typed, “repeat until beeper here.”

2.3 Data Collection and Analysis

We collected four primary forms of data to examine the impact of the online computing camp on engagement with CS, grasp of CS concepts, and overall satisfaction with the program: (a) recorded video sessions, (b) scores on the Computational Thinking Test (CTT), and (c) interviews with participants and their parents.

2.3.1 Recorded Video Session Data Collection and Analysis. Using seventeen recorded video sessions, we analyzed all of the interactions during the sessions to understand the engagement and learning of the participants. One author of this paper first transcribed the recorded sessions. Then both of the authors conducted open thematic coding. The first round of coding aimed to identify passages of text linked by a common theme. The authors discussed all of the misalignments, eventually establishing 100% agreement. 2.3.2 Computational Thinking Test.
The main CS concepts covered on CTT are the following: Basic directions and sequences (4 items), Loops using counts (4 items), Loops using “until” (4 items), simple If conditional (4 items), complex If/else conditional (4 items), While conditional (4 items), and simple functions (4 items) (Roman-Gonzalez et al., 2017). The questions covered three types of cognitive tasks: sequencing (14 items), completion (9 items), and debugging (5 items). Some of the questions assessed understanding of nesting.

2.3.3 Interview. We conducted semi-structured interviews with participants and their parents at the end of the program. The primary purpose of the interviews was to triangulate the findings regarding engagement, perceived learning, and satisfaction with the program. We asked the following questions: 1) Do you like programming? 2) Is the camp material challenging? Difficult? 3) Would you like to continue learning programming?

3. Results

This section begins by presenting the descriptive findings of the participants’ performance during and right after summer camp. Next, we present a qualitative analysis of their engagement with the camp material and instructors. All of the results were summarized around the research questions.

3.1 RQ1: Does the fully online, flipped computing summer camp facilitate the acquisition and development of computing concepts?

Table 1. Ali’s CTT Results

<table>
<thead>
<tr>
<th>Concept</th>
<th>Nesting</th>
<th>Yes (Correct)</th>
<th>No (Correct)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Basic directions and sequences</td>
<td>NA</td>
<td>4 out of 4</td>
<td></td>
</tr>
<tr>
<td>2. Loops using counts</td>
<td>NA</td>
<td>4 out of 4</td>
<td></td>
</tr>
<tr>
<td>3. Loops using “until”</td>
<td>2 out of 3</td>
<td>1 out of 1</td>
<td></td>
</tr>
<tr>
<td>4. Simple If conditional</td>
<td>1 out of 4</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>5. Complex If/else conditional</td>
<td>1 out of 4</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>6. While conditional</td>
<td>1 out of 4</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>7. Simple functions</td>
<td>1 out of 4</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>

3.1.1 CTT Result for Ali. Ali answered at least one question correctly for each concept, regardless of whether the question was in a nested format. He correctly answered almost all of the questions without nesting: basic sequences, Loops using counts, and Loops using “until.” However, he struggled with complex concepts and nested elements. Please see Table 1 for more details.

3.1.2 Field Notes through Recorded Videos and Interview with Ali. During the tutoring sessions, both the parents and the instructor asked Ali questions to ensure he understood code tracing, code building, code labeling, debugging, and basic concepts (e.g., “if we’re going to create a command, do you remember what it’s called in a snap?”). The code tracing questions required Ali to understand each line of code and decide which changes would obtain the desired outcome. Most of the time, he was able to answer correctly. However, the parents sometimes needed to repeat the question several times:

Mom: Tell me, where does this change need to go to? Look at these commands, and where does this change beeper count go to? Below what? Tell me.
Ali: Below pick-up beeper, is that right?
The questions about code building focused on the code featured in the assignments. Ali was able to follow the conversation and answered correctly. He asked for help two times during all of the sessions when answering questions about code building, especially ones related to loops and iteration.

The questions about code labeling were to teach Ali how to write clear code and label it meaningfully so that whoever might use it later could make sense of it. A few times throughout the program, the instructors asked students to name a piece of code. At the beginning of the intervention, Ali labeled the code pieces with real human names (e.g., “Cindy”). After explicit instruction, he named all of his code according to what it was or what its purpose was. We considered him a “verbose” code labeler because he used detailed names to label his code.

The students inevitably had to debug their code because that task is an important part of CS. The instructor helped with debugging by drawing attention to certain areas of the code where there might be a problem and often helped with very specific parts of more difficult code. Finally, the instructor asked some direct questions to check whether Ali understood the concepts. Ali was very good at those questions and accurate used CS jargon, an impressive feat given that he was spelling those terms:

Instructor: The conditionals are okay, and something else, he said something else: two things, two points he said. Do you remember the second one? What was it?
Ali: (spelling) Iterate.

In addition to these informal assessments during the tutoring sessions, Ali coded independently without the instructor and with little help from his parents (using the computer mouse or trackpad). His mom drew attention to his achievement during the interview:

Mom: Uh, tell him what you did this morning... We went through the whole of assignment one once again, and this time, Ali did it completely [by himself].
Instructor: Oh, excellent, Ali. There you go.”

3.2.1 Max’s CTT Results. Max answered 24 out of 28 questions correctly on CTT in ten minutes, quite faster than a typical student. He answered all questions about basic directions and sequences, Loops using counts, Loops using “until,” and While conditionals correctly. Although he missed only one question about complex If/else conditionals, he missed two questions about simple If conditionals. In addition, he correctly answered 3 out of 4 questions about simple functions.

3.2.2 Field Notes through Recorded Videos and Interview with Max. We conducted the same types of informal assessments with Max. Max’s parents asked him what to do to complete the assignment step-by-step. On one occasion, his mom told the researchers that he debugged a couple of mistakes and successfully built the code blocks. He also labeled the code correctly from the beginning. The instructor called him a concise labeler. Max learned the CS jargon quickly during the camp. He was able to talk about loops and conditionals early on and identify them as the main concepts:

Mom: You understand the fundamentals?
Max: Yes.
Mom: Like what?
Max: Loops, conditionals.

Some of the interview questions related to how much he learned. His mom reported that he did not have any problem understanding the material and that he could complete the assignments without help. Max’s dad talked about how Max caught errors in the code that he had missed.
himself. This outcome was impressive because the dad was a CS professional. Here is what the dad said about the mistake he made and how Max fixed it:

Dad: It was funny because he caught a problem I missed—turning left instead of turning right. And he created a turn right command instead of turning left a bunch of times.

3.2 RQ2: To what extent do participants enjoy and feel engaged in the computing instruction in the fully online, flipped summer computing camp?

3.2.1 Ali. Our observation data revealed that Ali engaged with the content and actively participated in the instruction. We recorded his engagement during the instruction, and his mom described his engagement with the course materials outside of synchronous sessions.

During the synchronous meetings, Ali appeared distracted many times. However, his responses to the questions about the content coming from the lead parent showed that he was focused and listening. He answered questions quickly without an extra reminder from his parents. His answers were mostly correct. Another indicator of Ali’s engagement during the synchronous meetings was his help-seeking behavior. When he did not know the answer to a question, he asked explicitly for help:

Mom: He’s saying, you tell me... the question is so hard.
Ali: I need help.

In addition, his mom explicitly checked his perceived comprehension of concepts during the meeting. He reassured his parents that he understood the material.

In contrast to the instances when Ali showed engagement with the content, he sometimes turned his attention to the food he was eating, a noise in the other room, or various other environmental elements. His parents reminded him to keep his hands on the board, look at the screen, focus, and sit down approximately 5 to 10 times a day.

Outside of the synchronous meetings, Ali watched the online materials with his parents. In addition, he willingly worked on the assignments. His mom expressed excitement about Ali completing an assignment without her help. While his mom was sharing her excitement, Ali also showed how he felt after completing an assignment or a piece of code: he spelled out “so happy.”

Based on the interview with Ali and his father, we concluded that Ali enjoyed working on CS-related projects. Ali said he liked the CS camp and thought it was “interesting to see how programming is done.” He admitted that he struggled at the beginning of the camp, but he did not say it was difficult at the end.

3.2.2 Max. Max attended two synchronous meetings. During those meetings, Max was on top of the content and able to build the code requested in the assignment. His parents never had to remind him to pay attention; he remained focused. During the interview, Max spelled that he found computing “difficult sometimes,” and he “liked it more than he thought he would.” He also stated that he would like to continue learning programming.

In addition, his mom said, “He’s very interested in programming. For each task, he likes to get finished. He feels good. He’ll stay here and sit with me until he feels like it’s finished. Then he gets up and celebrates.”

4. Discussion

Our study extends efforts to include students with ASD in CS fields (Begel et al. 2020; Israel et al. 2020). The main contribution of this study is the investigation of the impact of an online, flipped summer camp on the extent to which non-verbal autistic students engaged with and learned CS concepts. Two students with ASD participated in this study which took place in an online environment in a flipped course instructional design. Both of the students were non-verbal and communicated through spelling board during the camp with the help of their parents.
Our findings show that non-verbal students with ASD are highly engaged with the CS material. Their learning was no different from neurotypical students in the instructional setting of this study: (a) asynchronous online instruction, (b) 1:1 instruction online synchronous tutoring, and (c) communication support. This finding is in line with the previous studies (Lindsay & Hounsell, 2017; Lamptey et al., 2021). Max answered 24 out of 28 questions correctly on CTT. Both the parents and the instructor asked Ali and Max questions to ensure they understood code tracing, code building, code labeling, debugging, and basic concepts (e.g., “if we’re going to create a command, do you remember what it’s called in a snap?”). Both of them answered most of the questions correctly, and Ali asked for help occasionally (i.e., two times during the camp). In addition, in CTT test, Ali answered at least one question correctly for each concept, regardless of whether the question included nesting. However, he correctly answered almost all of the questions without nesting. These questions covered basic sequences, Loops using counts, and Loops using “until.” He struggled with complex concepts and nested elements. More research and efficient instructional materials are needed to teach the complex CS concepts (e.g., nested loops) than regular video-based lecturing, and scholars need to investigate the impact of these materials on learning. In future studies, we plan to compare the impact of code outlines and pseudocode text-based learning on students’ understanding of advanced computing concepts.

In addition, more validated instruments are in need of understanding how well individuals with ASD grasp CS concepts. There are few validated instruments available in the literature, but none of them were designed with students with ASD in mind. We used validated assessment by Roman-Gonzalez (2017). However, one of our students found it too long (28 questions total) and completed the test in two sessions. In the next iteration of this camp, we plan to use half of the questions covering the CS concepts outlined in the original test. Besides these, scholars need to investigate newly emerging alternative assessments (e.g., in-video questions or emoji-based surveys) to measure attitude toward CS learning, especially the ones that could be used in informal learning contexts.

5. Limitation

The limitations of this study call for future research. First, the generalizability of the findings is limited by the small sample size. Scholars need to evaluate the effectiveness of online computing teaching in a flipped format using a large sample size. Second, the participants were both over 20 years old and male. Future studies should include diverse participants so that the findings are more generalizable in terms of gender and age. Third, we focused on the participants’ grasp of computing concepts and practices and their engagement with the content. However, student-level factors have proven to influence performance in CSed, such as self-efficacy (Authors, 2019) and self-regulation (Lishinski et al. 2016). Scholars should consider including individual differences as factors related to the participation of individuals with ASD in CS activities. Fourth, both of participants use rapid prompting method which was questioned by a number of researchers and professional organizations. It was parents’ choice already and changing the way of communication was beyond the scope of our study. The main concern raised in the literature for rapid prompting is the uncertainty of authorship in communication. To overcome this issue, the research team watched the videos several times and revised the observation notes. Future studies should do similar intervention with students with ASD who uses some other communication method than rapid prompting.

6. Conclusions
Although CSed is gaining tremendous attention worldwide, research on individuals with cognitive disabilities in the context of CS is growing at a slower pace. As a result, there is a limited literature on CS education for students with disabilities, including with ASD. This study extends efforts to expose broader populations to CS and investigated the feasibility and effectiveness of using online learning in a flipped format to teach CS concepts and practices to individuals with ASD. We examined the participants’ learning of CS concepts and engagement during the camp. The findings suggest that online CS teaching studies featuring individuals with ASD can make positive contributions. With the right instruction and 1:1 support, individuals with ASD can successfully learn computing concepts and practices.

7. References


