Paper Title Student Perceptions of Computational Thinking (CT) Practices in a CT-Integrated Environmental Science Unit (Poster 12)

Author(s) Lexie Zhao, Northwestern University; Amanda Peel, Northwestern University; Uri J. Wilensky, Northwestern University; Michael S. Horn, Northwestern University

Session Title Computational Thinking and the Learner and Educator Experience (San Diego)

Session Type Poster Presentation

Presentation Date 4/21/2022

Presentation Location San Diego, California

Descriptors Computers and Learning, Curriculum in Classroom

Methodology Mixed Method

Unit Division C - Learning and Instruction

DOI https://doi.org/10.3102/1892662

Abstract:
Integrating computational thinking into science curriculum can deepen learning of science content and broaden participation in CT fields. This study uses a taxonomy of CT practices in STEM to frame the iterative design of an environmental science unit. The investigation focused on student perceptions of the CT-integrated unit to better inform the next iteration of this unit. On average, students' attitudes shifted negatively after the unit, but students differ in their perceptions of coding. From an iterative design perspective, we need to understand these perspectives to revise the lesson. Since students' perceptions shape their engagement and interest in continuing in computing-related endeavors, it is important to support comfort and confidence. Potential reasons for attitudinal shifts and implications are discussed.

Purpose:
Science and mathematics are becoming increasingly computational endeavors (Weintrop et al., 2016). The thoughtful application of computational tools and skill sets can deepen learning of mathematics and science content (Guzdial, 1994; National Research Council, 2011; Repenning et al., 2010; Sengupta et al., 2013; Author, 1995; Authors, 2014; Authors, 2006). CT can be effective to deliver better results when made an integral part of students' everyday science and mathematics learning (Authors, 2014).
One of the motivations for bringing CT into mathematics and science curriculum is to broaden participation and address longstanding issues of the underrepresentation of women and minority groups in computational fields (Authors, 2014). Given existing evidence that programming performance is related to confidence and attitudes to computing science (Jenson et al., 2016), there are several instruments that use Likert-scale questions to capture confidence and attitudinal shifts related to CS (Hoegh & Moskal 2009; Heersink & Moskal 2010). This research aims to capture the effect of CT-integrated science curriculum through the lens of students' perceptions about CT practices and computational tools. Specifically, we explore:

> *How did students' attitudes towards computational thinking and practices change after they participated in a CT-integrated environmental science unit and why?*

Framework:
In our study, the integration of CT and science content is realized through the application of Weintrop et al.'s (2016) Computational Thinking in Science and Math taxonomy. We used a revised version of the taxonomy of practices (CT-STEM 2.0 - Authors, 2021) to frame the design of an environmental science unit used in this study. The CT-STEM 2.0 taxonomy includes six practice categories: computational modeling, computational visualization, computational data, algorithms, programming, and computational problem-solving. The intent is to engage students in authentic science and math practices, have them use computational tools to understand science and math phenomena and solve science and math problems.

Methods and data sources:
<u>Participants and setting:</u>
The unit explored in this study was co-designed by an environmental science teacher from a midwestern urban high school and the second author. The unit engages students in learning about environmental systems through the CT-STEM 2.0 taxonomy's six categories. The unit

was taught during the 2020-2021 school year and was implemented remotely through the project website (blinded for review) and Google Classroom. The unit outline can be seen in Table 1 and explored here (link blinded). Student demographics can be seen in Table 2.

Data collection

We collected students' Likert responses (1=strongly disagree, 5=strongly agree) and short answers to open-ended questions from pre and post surveys, as well as three embedded surveys taken after lesson 1, lesson 5, and lesson 7. In general, Likert items attended to students' enjoyment and confidence regarding taxonomy CT practices and computational tools. Open ended questions focused on what students found challenging, what they liked from the lessons, and what they learned.

Data analysis

We focus on quantitative analysis from pre-post surveys, three embedded surveys, and qualitative analysis from pre-post surveys. We started with 30 pre-unit responses, and we removed 4 from quantitative analysis due to missing data points, resulting in $N = 26$, and we removed 5 from qualitative analysis due to missing data points, resulting in $N= 25$. Due to the nature of our small sample, we used Wilcoxon Signed Rank test for nonparametric data (Laerd Statistics, 2015) for pre-post Likert responses. Effect sizes for each comparison were calculated according to Field (2013) and Hattie (2009). For embedded survey Likert responses, we used Friedman's Test and Fisher's Least Significant Difference Post Hoc Test with the Bonferroni adjustment (Laerd Statistics, 2015). For open-ended questions, we used deductive coding and inductive coding for trends within the four selected questions.

Findings

Pre-Post Differences

While students' confidence with CT-STEM 2.0 taxonomy practices did not change significantly, during the implementation, their enjoyment significantly decreased from pre to post for all CT-STEM 2.0 taxonomy practices except programming (Figures 1 & 2; Table 3). The scores for programming were low before the unit in comparison to other practices, which may explain the lack of significance. Students were also less motivated to include computational tools in their future learning and careers after the unit learning (Figure 3). For instance, they were less interested and confident in careers that involve the use of computational tools such as a scientist, engineer, or computer programmer one day. Out of twelve pre survey questions, there are eleven questions where students' mean response scores were above three on a scale of one to five, indicating that on average they held neutral to positive views about including computational tools in the future before the unit. However, after the unit, mean response scores dropped below three, indicating their attitude shifted from neutral to slightly negative (Figure 3).

Embedded Assessments

First Lessons were science lessons without CT. Middle lessons engaged students in CT-STEM 2.0 taxonomy practices related to exploring computational models and computational data. Last lessons were focused on programming, algorithms, and problem solving with CT. Students' engagement in lessons did not differ significantly throughout the unit (Figure 4, Table 9). In general, first lessons had more positive responses than later ones (Figure 4, Table 6 & 7 & 8 & 9 & 10). Students enjoyed the first lesson more than the last lessons (Figure 4, Table 7). Students felt they successfully learned better in the first unit compared to the last lessons in the unit (Figure 4, Table 8). Students also found middle and last lessons more challenging and stressful than the first lesson (Figure 4, Table 6).

Student Open-Ended Responses

To better understand why the students' self-reported assessment decreased over the unit, we analyzed their post-unit responses. We focused on four open-ended questions: 1) what did you learn? 2) What did you enjoy? 3) What was helpful? 4) What did you not like?

Self-reported learning. When asked what students learned from this unit, the whole class reported learning about *coding, algorithms, and modeling*.

> *"I learned that by using computation models and algorithms/simulations it can help us learn more about the real world and collect data that could be used to see what could happen in real life. They can also help us understand these scenarios more and get a better understanding of them."*
> *"I learned how to actually code, but to a deeper extent than the other when I would use different websites growing up. I also got to learn how an algorithm really does play a big role and how it's important to form one."*

In comparison, *science content* was mentioned only by two students.

> *"Many things, from learning to code in order to learning about different kinds of systems and how they affect the world as a whole."*

This indicates that CT-related content left a strong impression on students, and students connected that knowledge to their life experiences and epistemology.

Students who liked coding: Although the quantitative analysis shows a decreasing trend in interests, we discovered a group of students who enjoyed learning coding and found it helpful. When asked what they enjoyed in this unit, ten students out of twenty-five reported the coding lessons. Specifically, most of them mentioned using NetLogo (a computational modeling environment; Wilensky, 1999a) to make turtles move and change colors.

> *"I enjoyed using the turtle model and programming them to do whatever I wanted."*
> *"I enjoyed making all my turtles Yellow and making color designs in my model."*

This group of students were drawn to the constructionist element of the coding activity that gave them agency. Additionally, four students also mentioned that the turtle activity helped them learn science content.

> *"Lesson 7 and actually coding made me more interested in the topic."*
> *"Lesson 7 when we got to create our own codes with the turtles."*

Students who disliked coding: Although students reported learning about CT-related knowledge, some did not necessarily like it. Eleven out of twenty-six students said they disliked coding. The challenging content put some students under stress.

> *"I didn't like the coding part to be honest because I suck at it and it stressed me out a lot."*
> *"We would move at a very fast pace for me sometimes and once we got to the coding it only got harder to understand."*

These findings may explain the sudden increase of stressfulness reported in lessons that contained coding elements in our quantitative analysis and align with the decreasing trend in enjoyment throughout the CT learning process.

Students who liked group work: Nine students regarded collaborative learning as helpful and enjoyable.

> *"...I enjoyed working with my group."*
> *"What I enjoyed was the breakout rooms because we helped each other."*

When asked what activities in this unit helped them learn the science content, seven students reported that the collaborative learning process helped.

> *"The group work made me understand things from a different point of view and made me understand it better."*
> *"Working with groups to break down the model and give each other ideas."*

Out of the eleven students who expressed negative perceptions about coding, seven mentioned group work as helpful and enjoyable.

Other students: Students mentioned other factors, such as remote learning and the clarity of instructions, that may contribute to their negative attitudinal changes.

> *"I didn't like how it felt like we were forced to understand this throughout the screen because I believe it would have been better through person."*
> *"I didn't like how vague the directions were at times because it led to a lot of confusion."*

Discussion & Implications

It is important to understand successes and challenges associated with student learning and student perspectives during our iterative co-design and revision of CT-integrated science units. While other studies provide a general account of students' perceptions of computational thinking (Tang et al., 2020), this study provides a more nuanced exploration of students' perceptions of specific CT-STEM 2.0 taxonomy practices. Our preliminary results suggest that students need more support in algorithm, programming, and problem solving practices to feel successful and confident. As students engaged in lessons relating to computational modeling and data, many of them felt these lessons were more challenging than science-only lessons. During the final lesson which introduces algorithms, programming, and problem-solving, most students felt more challenged and stressed. Across different CT-science practices, students on average felt least confident and happy about programming, which aligns with other literature about student perceptions of coding and programming (e.g., Begel et al., 2007).

While the unit was contextualized in an environmental science problem relating to farming and sustainability, our findings suggest that contextualizing CT practices with real-world problems alone may not be enough to attract all students. In this work, even though students reported deeper understandings of CT, they were less interested in those pathways when compared to the beginning of the unit. Since students' perceptions of programming and their experiences shape their learning abilities and impact their career interests (Biggers et al., 2008; Weintrop et al., 2016), it is imperative to avoid negative experiences with programming to broaden participation in computing-related STEM fields.

Although most students reported negative perceptions of CT practices, some students enjoyed programming. This suggests different students need different levels of scaffolding and types of support when it comes to learning CT-integrated science content. For those who are less comfortable or familiar with coding, collaborative learning processes may help students feel less overwhelmed and intimidated as they solve problems, code, and create algorithms as a group and help each other.

As reported in some student responses, remote learning could have impacted students' perceptions of the unit. It hindered student-teacher interactions because all students had their videos turned off in Google Meetings and students rarely shared ideas or communicated. It is possible this lack of interaction contributed to frustration with coding since the teacher could not measure progress, frustration, or learning through blank screens.

References

Author, 1995

Author, 1999

Authors, 2006

Authors, 2014

Authors, 2021

Authors, 2021

Begel, A., & Klopfer, E. (2007). Starlogo TNG: An introduction to game development. *Journal of E-Learning*, *53*(2007), 146.

Bell, P., Van Horne, K., & Cheng, B. H. (2017). Designing learning environments for equitable disciplinary identification.

Biggers, M., Brauer, A., & Yilmaz, T. (2008). Student perceptions of computer science: a retention study comparing graduating seniors with cs leavers. *Acm sigcse bulletin*, *40*(1), 402-406.

Field, A. (2013). *Discovering statistics using IBM SPSS statistics.* sage.

Guzdial, M. (1994). Software-realized scaffolding to facilitate programming for science learning. *Interactive learning environments*, *4*(1), 001-044.

Heersink, D., & Moskal, B. M. (2010, March). Measuring high school students' attitudes toward computing. *In Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 446-450).

Hoegh, A., & Moskal, B. M. (2009, October). Examining science and engineering students' attitudes toward computer science. In 2009 39th IEEE Frontiers in Education Conference (pp. 1-6). IEEE.

Jenson, J., & Droumeva, M. (2016). Exploring media literacy and computational thinking: A game maker curriculum study. *Electronic Journal of e-Learning*, 14(2), pp 111-121.

Jiang, S., & Wong, G. K. (2017, December). Assessing primary school students' intrinsic motivation of computational thinking. In *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)* (pp. 469-474). IEEE.

National Research Council. (2011). *Learning science through computer games and simulations*. National Academies Press.

Repenning, A., Webb, D., & Ioannidou, A. (2010, March). Scalable game design and the development of a checklist for getting computational thinking into public schools. In Proceedings of the 41st ACM technical symposium on Computer science education (pp. 265-269).

Säde, M., Suviste, R., Luik, P., Tõnisson, E., & Lepp, M. (2019, February). Factors That Influence Students' Motivation and Perception of Studying Computer Science. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (pp. 873-878).

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. Education and Information Technologies, 18(2), 351-380.

Hattie, J. (2009). Visible learning. London, UK: Routledge.
Laerd Statistics. (2015). Statistical tutorials and software guides. Retrieved from https://statistics.laerd.com/
Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, *148*, 103798.
Jona, K., Wilensky, U., Trouille, L., Horn, M. S., Orton, K., Weintrop, D., & Beheshti, E. (2014, January). Embedding computational thinking in science, technology, engineering, and math (CT-STEM). In *future directions in computer science education summit meeting, Orlando, FL*.
Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127-147.
Weintrop, D., & Wilensky, U. (2016). Bringing blocks-based programming into high school computer science classrooms. In *Annual Meeting of the American Educational Research Association (AERA). Washington DC, USA*.
Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
Yoon, S. A., Goh, S. E., & Park, M. (2018). Teaching and learning about complex systems in K–12 science education: A review of empirical studies 1995–2015. Review of Educational Research, 88(2), 285-325.

Appendix:

Table 1. CT-integrated Environmental Science Unit Outline

Lesson	**Description**	**CT-Science Practices**
1. Intro to Systems	Students learn about common systems and how a system is defined *First embedded assessment	Science Only
2. Drawing Systems Diagrams	Students learn about different system types and how to represent systems and dynamics through hand drawing	Science Only
3. Intro to Models	Students learn about the different types of models and are introduced to computational models	Computational Modeling, Computational Data, Computational Visualization, Programming
4. Energy in a System	Students learn about types of energy and energy transfer and transformation in systems using computational modeling	Computational Modeling, Computational Visualization

5.System Equilibria and Complexity	Students explore complexity and equilibrium through computational models *Second embedded assessment	Computational Modeling, Computational Data, Computational Visualization
6. Feedback Mechanisms	Students use and create computational models to learn about positive and negative feedback	Computational Modeling, Computational Data, Computational Visualization, Programming
7. Applying Models to Solving Real World Problems	Students use algorithms, programming, and computational models to solve a real world farming problem *Last embedded assessment	Computational Modeling, Algorithm, Computational Visualization, Programming, Computational Problem-Solving

Table 2. Student Demographics

Student Category	**Number**	**Percentage**
Total Students	30	N/A
Female	19	63.3
Male	11	36.7
Non-binary	1	3.3
American Indian or Alaskan Native	1	3.3
Asian, White	1	3.3
Black or African American	2	6.7
Black or African American, Hispanic or Latino, White	1	3.3
Hispanic or Latino	23	76.7
Middle Eastern or North African	1	3.3
White	2	6.7

Figure 1. Students' Pre-post Responses on Enjoyment

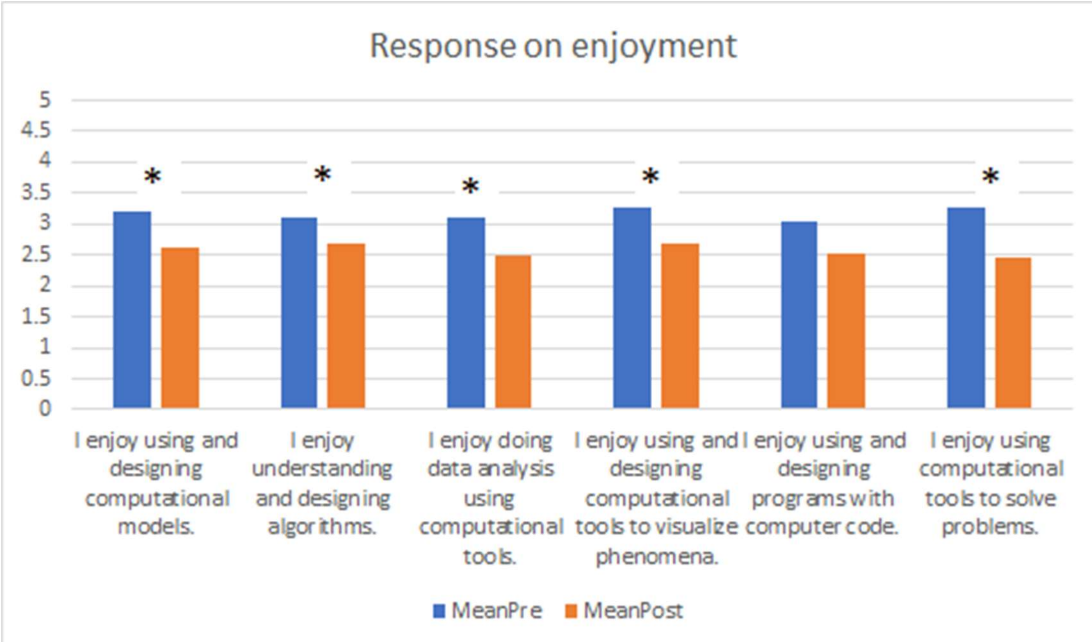

Figure 2. Students' Pre-post Responses on Confidence

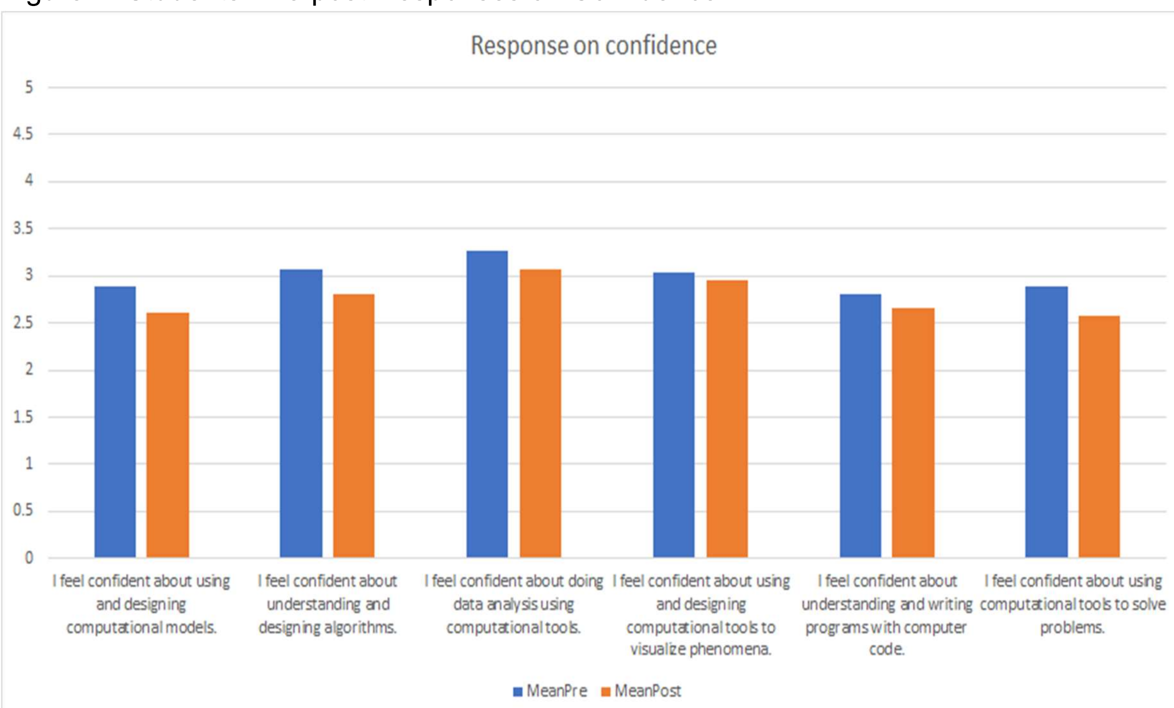

Table 3. Detailed Quantitative Analysis Results for Students' Pre-post Responses - Taxonomy*

CT-Sci Practice	Perception	Pre Mean	Post Mean	p Value	Effect Size r
	Confidence	2.884615	2.615385	0.227264	0.2367977

Computational Modeling	Enjoyment	3.192308	2.615385	**0.025011**	**0.4395428**
Algorithms	Confidence	3.076923	2.807692	0.282822	0.2106292
	Enjoyment	3.115385	2.692308	**0.043285**	**0.3963404**
Computational Data	Confidence	3.269231	3.076923	0.471041	0.1413579
	Enjoyment	3.115385	2.5	**0.025093**	**0.4392933**
Computational Visualization	Confidence	3.038462	2.961538	0.682869	0.08012345
	Enjoyment	3.269231	2.692308	**0.044377**	**0.3942941**
Programming	Confidence	2.807692	2.653846	0.539255	0.1204037
	Enjoyment	3.038462	2.538462	0.054918	0.3764503
Computational Problem-Solving	Confidence	2.884615	2.576923	0.115947	0.3082975
	Enjoyment	3.269231	2.461538	0.001204	0.6350223

*r between 0.1 and 0.25 = small effect, 0.24–0.36 = moderate effect, and 0.37 and higher = large effect.

Figure 3. Students' Pre-post Responses on Future Involvement of Computational Tools in Learning and Career

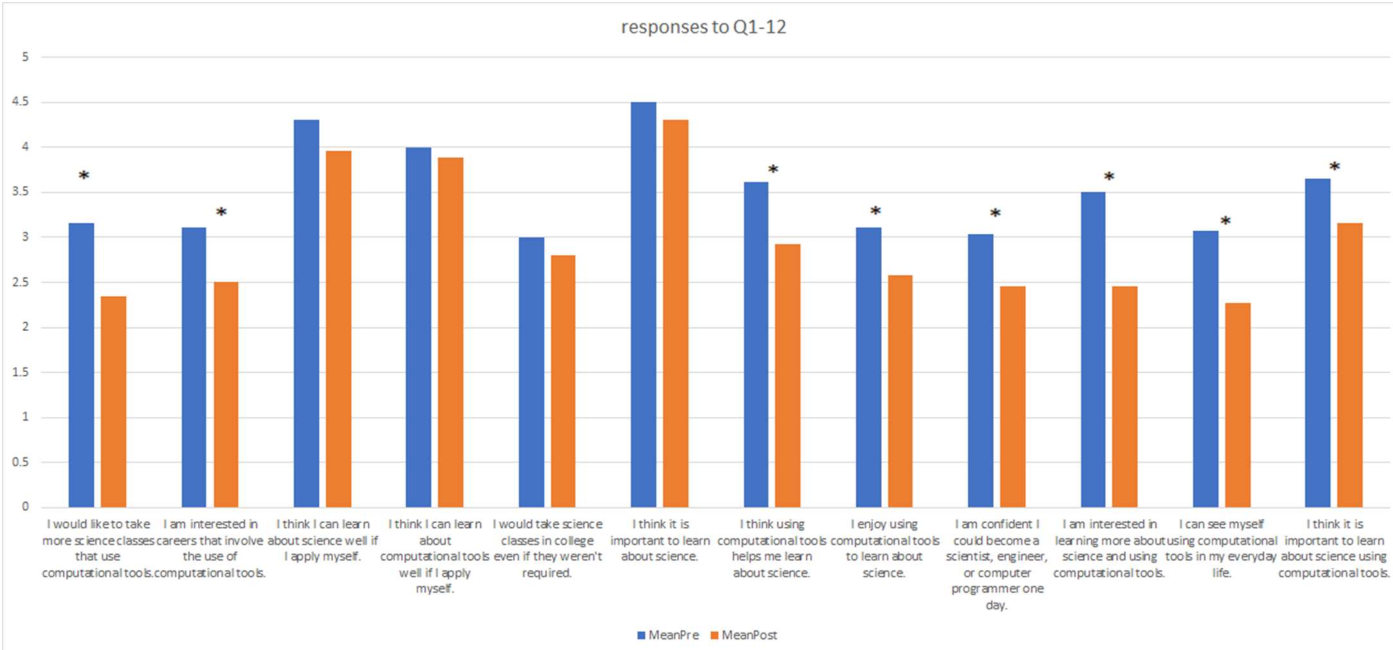

Figure 4. Students' Responses for Embedded Lessons

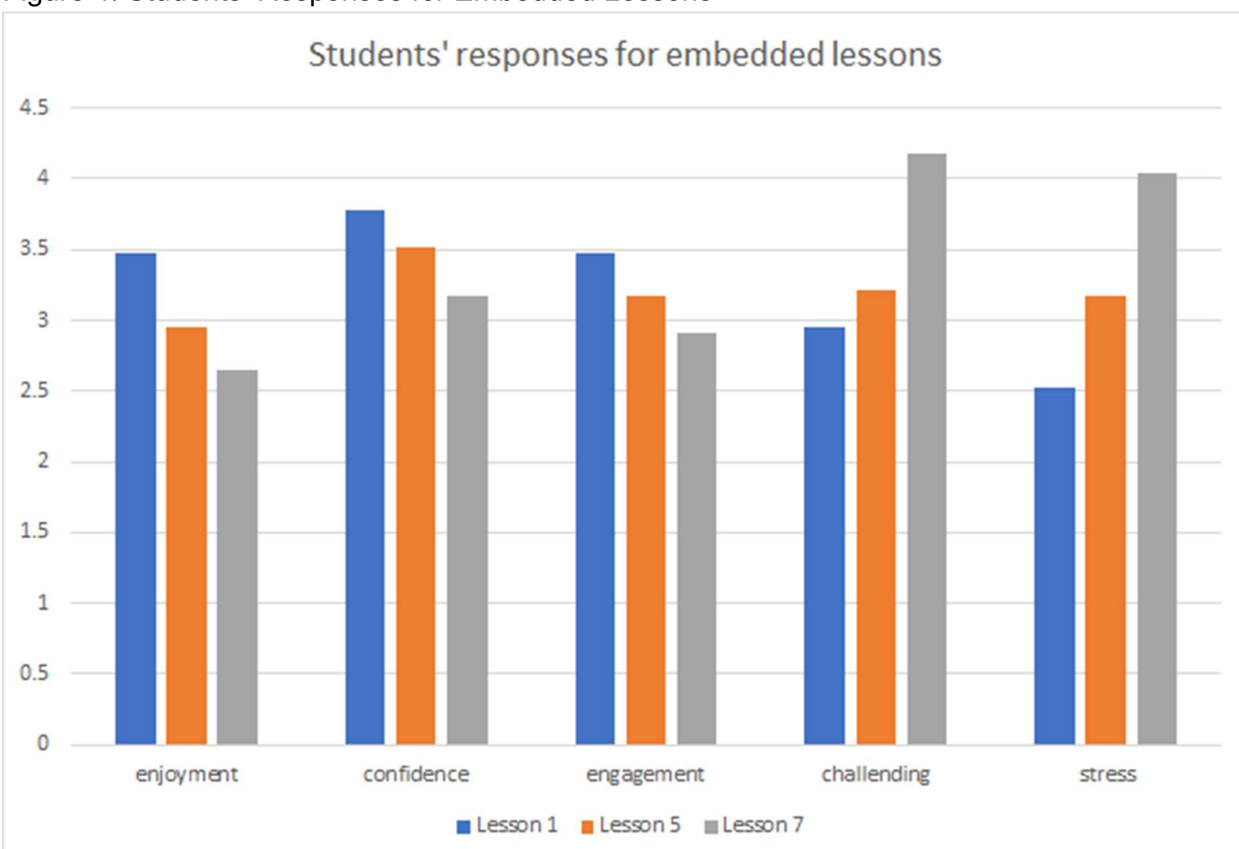

Table 4. Detailed Quantitative Analysis Results for Students' Pre-post Responses on Future Involvement of Computational Tools in Learning and Career

Likert Item	Pre Mean	Post Mean	p value	Effect Size r
I would like to take more science classes that use computational tools.	3.153846	2.346154	**0.003288**	**0.5764698**
I am interested in careers that involve the use of computational tools.	3.115385	2.5	**0.011345**	**0.4965461**
I think I can learn about science well if I apply myself.	4.307692	3.961538	0.097172	0.3253021
I think I can learn about computational tools well if I apply myself.	4	3.884615	0.482056	0.1378696
I would take science classes in college even if they weren't required.	3	2.807692	0.286924	0.2088428
I think it is important to learn about science.	4.5	4.307692	0.307514	0.2001271
I think using computational tools helps me learn about science.	3.615385	2.923077	**0.01453**	**0.4792854**
I enjoy using computational tools to learn about science.	3.115385	2.576923	**0.022167**	**0.4486146**
I am confident I could become a scientist, engineer, or computer programmer one day.	3.038462	2.461538	**0.001976**	**0.6067342**
I am interested in learning more about science and using computational tools.	3.5	2.461538	**0.000846**	**0.6544911**
I can see myself using computational tools in my everyday life.	3.076923	2.269231	**0.001174**	**0.6364193**
I think it is important to learn about science using computational tools.	3.653846	3.153846	**0.013141**	**0.4863511**

*r between 0.1 and 0.25 = small effect, 0.24–0.36 = moderate effect, and 0.37 and higher = large effect.

Table 5. Students' Perceptions on Coding and Collaborative Learning

	Positive about coding	Negative about coding	Coding not mentioned
Positive about collaborative learning	1	7	1
Negative about collaborative learning	2	0	0
Collaborative learning not mentioned	8	4	0

(Note: There are 2 students who do not fit into this table. They seem to have mixed feelings about coding or collaborative learning, e.g. enjoys discussion as a class but not the group work or enjoys turtle activity but not coding in depth.)

Table 6. Stress Embedded Assessment Comparison ($F = 28.25103$, $p = 1.282652e{-08}$, $LSD = 10.10327$)

Lesson #	Mean	Rank Sum	Group*
1	2.521739	31	a
5	3.173913	45.5	b
7	4.043478	61.5	c

*Treatments of the same letter are not significantly different.

Table 7. Enjoyment Embedded Assessment Comparison ($F = 5.339983$, $p = 0.008389529$, $LSD = 13.01445$)

Lesson #	Mean	Rank Sum	Group*
1	3.478261	55	a
5	2.956522	45	ab
7	2.652174	38	b

*Treatments of the same letter are not significantly different.

Table 8. Learning Success Embedded Assessment Comparison ($F = 7.143992$, $p = 0.002056933$, $LSD = 11.90715$)

Lesson #	Mean	Rank Sum	Group*

1	3.782609	55	a
5	3.521739	45	ab
7	3.173913	37	b

*Treatments of the same letter are not significantly different.

Table 9. Engagement Embedded Assessment Comparison (F = 2.401015, p = 0.102406, LSD = 114.89605)

Lesson #	Mean	Rank Sum	Group*
1	3.478261	52	a
5	3.173913	47	a
7	2.913043	39	a

*Treatments of the same letter are not significantly different.

Table 10. Challenging Embedded Assessment Comparison (F = 9.760963, p = 0.0003101976, LSD = 12.79625)

Lesson #	Mean	Rank Sum	Group*
1	2.956522	38	a
5	3.217391	41	b
7	4.173913	59	b

*Treatments of the same letter are not significantly different.