Algorithms in Art and Code: How Teaching Embodied Artmaking Procedures Can Stimulate Analytical Thinking in Art Crafting and Computer Programming

Jacqueline Elise Bruen
Virginia Tech
Computer Science
Blacksburg, Virginia, USA
jacquelineb@vt.edu

Myounghoon Jeon
Virginia Tech
Industrial and Systems Engineering
Blacksburg, Virginia, USA
myounghoonjeon@vt.edu

Abstract

People have pointed to a connection between the creative arts and computing. In the present longitudinal pilot study we taught six programmers and six non-programmers how to read and write written crochet patterns with or without the accompanying crochet gestures. Half of programmers (three participants) and non-programmers (three participants) were taught with the gestures, while the other halves were not. Over two weeks we individually taught participants crochet during three separate 30 minute sessions. In a fourth session, we tested participants on crochet and elementary programming and algorithms. Test results showed that programmers and non-programmers performed better on average on both tests when they learned with gestures. We interviewed all participants afterwards; programmers provided examples of how crochet demonstrated elementary programming ideas, while non-programmers described what they thought about programming. Our empirical study provides evidence of embodied cognition and offers contributions towards developing novel teaching methods in computer science.

CCS Concepts

• **Human-centered computing** → **Empirical studies in HCI**; *Gestural input*; • **Applied computing** → *Arts and humanities.*

Keywords

Embodied Cognition, Computer Science Education, Learning, Empirical Study

ACM Reference Format:
Jacqueline Elise Bruen and Myounghoon Jeon. 2025. Algorithms in Art and Code: How Teaching Embodied Artmaking Procedures Can Stimulate Analytical Thinking in Art Crafting and Computer Programming. In *Creativity and Cognition (C&C '25), June 23–25, 2025, Virtual, United Kingdom.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3698061.3734412

C&C '25, Virtual, United Kingdom

ACM ISBN 979-8-4007-1289-0/25/06
https://doi.org/10.1145/3698061.3734412

1 Introduction

Those working in the field of computer science (CS) can attest to the creativity required in their field [22]; elegant solutions in computing are highly revered and studied [5, 21]. While the creativity of CS may be obvious to computer scientists, it can elude the uninitiated [26, 27]. Ideas of what CS is have been built up over time and have left impressions on the general public [1, 30]. Individuals who exhibit the qualities of a great computer scientist may not see how their skills translate to computing.

While researchers have made efforts to promote CS from an art-based perspective, these projects often pertain to using CS to enhance, change, or build art [7, 20, 25]. These projects divide CS and art such that computing serves as the cause and art serves as the effect. Little has been done to utilize art as both the cause and the effect in demonstrating analogous qualities between the two fields.

Showcasing the similarities between art-making and computing may provide opportunities for building computing confidence [29] in the many individuals who already enjoy practicing art. Our approach to using art as a catalyst to drive new, creative perspectives to the tech sector is novel in its focus in highlighting the procedural aspects of creating art, and framing CS techniques as a new flavor of art-making strategies.

Crochet is a popular fiber art whereby the crocheter can design and create physical items by following a pattern. A crochet pattern communicates conditional procedures and repetitions. Thus, crochet patterns can exhibit similar characteristics to computer programs. The similarities between the two led us to design the present study. The present work has both practical and theoretical implications. From a practical perspective, our findings can motivate novel teaching methods in CS and similar fields. This study offers two new approaches: 1) artmaking is used as a medium to practice analytical thinking, and 2) physical interaction is used to support this thinking. The concept of embodied cognition [28, 31] suggests that cognition takes place in the central, perceptual, and motor systems of the body, and that our physical interactions can lend themselves towards cognitive objectives such as learning. Empirical studies on learning tasks have supported these claims [10, 19]. Thus, we utilized embodied cognition in the present study by incorporating physical interaction in learning as one of our two independent variables.

Our work's theoretical contribution comes as an application of Holyoak and Thagard's multiconstraint theory [16, 17], which posits that similarity, structure, and purpose constrain our ability to

develop and understand analogies. Our study design seeks to draw out from participants through tests and interviews whether these constraints are upheld in building analogies between computing and crochet.

In this pilot study, we taught participants with and without programming experience how to read, write, and interpret crochet patterns using two different learning methods (with or without gestures). We then tested them on crochet patterns and simple programming and algorithms in two separate tests. Afterwards, we interviewed the participants. Our research questions were as follows:

- RQ1: How do programmers perform on the crochet test compared to non-programmers?
- RQ2: How does learning the crochet gestures affect participants' performance on the crochet test or the programming test?
- RQ3: How do participants find similarities and differences between crochet and programming?
- RQ4: How are participants' impressions of programming after being introduced to the programming-like art practice of crochet?
- RQ5: How do participants' perceive their performance on the crochet test and the programming test, compared to their actual performance?

2 Related Work

2.1 Crochet

Crochet is a fiber art where the artist repeats a series of stitches to create physical objects. There are six foundational stitches which can be combined to make increasingly intricate designs. Crochet objects can be made by following a pattern. Patterns use abbreviations, syntax, and occasionally written language to explain stitch arrangements. In the present study, we taught all participants the abbreviations and syntax found in crochet patterns.

2.2 Analogies in Learning

Scientists make use of a variety of analogies to explain ideas in their fields [8, 9]. Researchers have applied analogies to teach topics in science [3]. While some have used analogies to promote recall or clarify an idea [11, 14], others have claimed that analogies can serve as a medium for discourse on complex topics [15]. Work in the field of psychology on building analogies, specifically multiconstraint theory [16, 17], outlined three requirements in building a successful analogy. These included similarity between the compared elements, consistent parallels in the elements' roles, and clear goals in developing the analogy. Our analogies meet these three criteria and thus, we used them to investigate how teaching crochet can help build computational thinking skills.

2.3 Embodied Cognition

Embodied cognition is a theory which posits that thinking is an activity served by the entire body in the context of a wider, interactive world. Research applying the theory has demonstrated the value of utilizing multiple modalities of perception to support learning [19, 23, 24]. The use of hand gestures, specifically, to convey information [13] has been used as a learning tool in an array of contexts [12, 33]. In our application of embodied cognition, we taught half of participants the physical gestures to crochet. While it was not necessary for participants to learn the gestures to master the written material, the literature in embodied cognition suggested that incorporating this modality could improve learning.

3 Method

3.1 Participants

Twelve participants (six male, six female) were recruited for the present pilot study. Six had taken at least one computer science class and knew how to write code in a programming language without help. All participants started with no crochet experience.

Of the six participants **with** programming experience, half (three) learned crochet patterns with gestures and written materials, while the other half (three) learned crochet using only written materials. The six participants **without** programming experience were assigned to the two conditions in the same way.

3.2 Experimental Design

A 2x2 between-subjects design was applied to answer our research questions. Our two independent variables were 1) crochet learning method, and 2) programming experience. Those with programming experience had taken at least one programming course and knew how to code in any language, unaided. Our dependent variables included the results of the crochet pattern test and programming test along with the interview data.

Each participant met individually with the researcher on four separate occasions. These meetings occurred no less than 24 hours apart for all participants and lasted approximately 30 minutes each.

3.3 Procedure

At the start of the first meeting, individuals were given an informed consent document approved by the university's IRB. Eligible individuals then read this document and signed it if they consented to be a participant. In the first two meetings, participants had a lesson (10 minutes) on the syntax and rules of written crochet patterns and were introduced to the abbreviations and symbols for three foundational crochet stitches. After the lesson, the participant was quizzed on the material with three questions (5 minutes). If the participant answered incorrectly, the researcher would demonstrate how to answer the question. Participants then reviewed flashcards on the material they had learned. After flashcards, participants assigned to the groups which only learned with the written materials were finished with the session. Participants in the groups assigned to learn with written materials and gestures learned two crochet stitches in the first two sessions. Participants who learned how to make these crochet stitches were provided with the materials, and sat next to the researcher as they demonstrated how to make each stitch. All participants who learned crochet stitches were successful in replicating each stitch unaided at least one time.

In the third session, participants reviewed all flashcards. Participants were then asked questions about three written crochet patterns that included all the information they had previously learned. After answering the questions, participants assigned to the gesture

groups were asked to recreate a simple crochet pattern alongside the researcher. Participants assigned to the groups that learned using only the written material were asked to write their own crochet pattern and explain it.

In the final session, participants took two tests. The first test was on crochet patterns (10 minutes) and the second test on simple computer programming and algorithms (10 minutes). Participants were provided pen and paper to work out the problems. All participants completed both tests within the allotted time. The crochet test in Appendix A.1 was developed by the researchers and consisted of nine multiple choice questions which only tested participants on the application of material from the written lessons. Note that these lessons were presented to all participants. The programming test consisted of nine (eight multiple choice and one multiple selection) mock questions [6] from the AP Computer Science Principles Exam [4]. This exam is a standardized test taken by high school students to earn college credit for an introductory CS course. Questions on this exam are written in a mock programming language, thus our programming test was also written in this mock language. This was advantageous for the present study, because writing the test in a particular language would risk capturing experienced CS participants' familiarity with the language rather than their problem-solving skills.

After participants finished the tests, they completed a semi-structured interview (10 minutes). Interview questions are in Appendix A.2 and covered topics including participants' perception of their test performance, their interest in the test topics, their confidence in expanding their knowledge of the test topics, and what similarities or differences they noticed between the tests.

3.4 Analysis

3.4.1 Crochet and Programming Tests. To analyze the quantitative test results, we ran descriptive statistics and a two-way multivariate analysis of variance (MANOVA) to determine whether there are overall effects on multiple dependent measures as a whole.

3.4.2 Interviews. We conducted a thematic analysis using the interview transcripts. This analysis consisted of five steps: preliminary note-taking, coding, generating themes, defining themes, and merging independent per-group analyses. Two researchers on the team independently performed the first four steps; artifacts from each step were separated by participant group. In the final stage of the analysis, a third researcher read through the two independent analyses and coalesced them by finding shared codes and notes. Theme definitions were revised to account for the input of both independent analyses.

4 Results

4.1 Crochet and Programming Tests

Table 1 displays the test scores and groups for all participants.

No statistically significant results were found from our MANOVA tests. Table 2 shows average scores computed within the groups; this table shows that scores on the crochet and programming tests were on average higher for groups which learned crochet with gestures compared to groups which only learned using the written lessons. Programmers tended to benefit more on both tests from adding gestures than non-programmers.

ID	Prog Experience	Learning Type	Crochet	Prog
1	Exp	Without gestures	88.89%	88.89%
2	Exp	Without gestures	11.11%	22.22%
3	Exp	Without gestures	100.00%	88.89%
4	Exp	With gestures	88.89%	77.78%
5	Exp	With gestures	77.78%	66.67%
6	Exp	With gestures	88.89%	100.00%
7	NoExp	With gestures	100.00%	55.56%
8	NoExp	Without gestures	66.67%	66.67%
9	NoExp	Without gestures	55.56%	55.56%
10	NoExp	Without gestures	77.78%	77.78%
11	NoExp	With gestures	88.89%	88.89%
12	NoExp	With gestures	44.44%	66.67%

Table 1: Test scores of each participant

Participant Group	Crochet Test	Programming Test
Computing experience, Without crochet gestures	66.67%	66.67%
Computing experience, With crochet gestures	85.19%	81.48%
No computing experience, Without crochet gestures	74.04%	66.67%
No computing experience, With crochet gestures	77.78%	70.37%

Table 2: Average scores on both tests for each participant group

4.2 Interviews

Table 3 displays the themes and definitions collected from our thematic analysis. We developed themes for each condition group.

5 Discussion

Through our investigation we were able to situate quantitative test results in a greater context through interviews. Participants' beliefs about programming, art, and their abilities were indicative of their will to continue learning programming.

5.1 Crochet and Programming Tests

Test results revealed two trends that merit further investigation. One of these was that adding gestures on average improved the scores of both programmers and non-programmers on both tests, thereby positively answering RQ2. We also saw a potential interaction effect between our two independent variables to help us answer RQ1: programmers tended to benefit more (had higher score increases on both tests) from using gestures to support their learning than non-programmers. These results align with past applications of embodied cognition [28, 31] in using gestures to support learning [32].

Participant Group	Themes and their Definitions
Computing experience, Without crochet gestures	• **Crochet skills are a subset of programming skills**: Participants found that the crochet skills they learned were similar to those of entry level programming. • **Programming takes more lessons to get the basics than crochet**: Participants shared that it would take more lessons to learn the basics of programming than it would for crochet. • **Programming and crochet both clearly outline what needs to happen next**: Participants found that crochet and programming demonstrated following procedures and instructions.
Computing experience, With crochet gestures	• **Programmers enjoyed the formality of crochet rules**: Those in this group shared that they enjoyed the specificity of the rules in crochet. • **Repetitions and stitches in crochet are similar to looping and functions in programming, respectively**: Participants made direct connections between specific concepts in programming and those they learned in crochet. • **Crochet vocabulary is limited, but programming knowledge is limitless**: Participants shared that the scope of what there is to learn in crochet is limited compared to programming.
No computing experience, Without crochet gestures	• **Crochet and programming have cause and effect relationships**: Participants in this group shared that the crochet and programming tests both included questions about the causes of results and the effects of procedures. • **Would continue programming for work reasons**: Participants in this group shared that they could see themselves learning programming for potential career advancement.
No computing Experience, With crochet gestures	• **Crochet is like algebra and programming is like logic**: Participants shared that both tests reminded them of math, specifically crochet resembled arithmetic and algebra, while programming resembled logic. • **Differentiating their skill sets from programmers**: Participants in this group emphasized the differences between programmers' skill sets and their own, implying that the skill sets were mutually exclusive.

Table 3: Participant groups and their themes and definitions

5.2 Programmers Describe Specific Examples of How Crochet and Programming Relate

In our interviews with programmers we found specific examples of how programming and crochet are similar. P6, a teaching assistant for an introductory programming class shared, "...I saw a lot of similarities. So first, if you want to repeat an instruction or a set of instructions [in crochet] *n* number of times it's very similar to doing for loops in programming... there are a set of instructions like single crochet, double crochet. We have similar instructions in programming..." P1 shared that they used the same strategies while taking the two tests, "so they both were... to count and find the answers. So they had pretty, like, big similarity... like how we use arithmetic operators when we learn in any programming language, it was just like that." P2 highlighted the procedural aspects of both subjects: "I think the process itself... there's always math involved... So I think math is the element that has a similarity in both." These quotes from programmers align with the multiconstraint theory [16, 17] for building analogies in that participants described how crochet and programming were similar in form and usage. Thus, as the theory predicted, participants were able to fill in the gaps of dissimilarity to recognize how the two disparate domains could be used in tandem to support their success on the tests. Works such as [2, 18] have also tested multiconstraint theory to support learning tasks across domains with positive results.

Programmers explained that while crochet has similarities to programming, these similarities are just a subset of programming. P6 stated, "So the crochet test... the first difference that I saw was it was limited to some vocabulary while the programming test had no limits... Other than that, I didn't see much difference. It was, it is a kind of algorithm, crochet, that you repeat some number of times, so it's very similar to programming or algorithms." P3 compared their experience learning the two subjects by saying, "If I need to compare then I think crochet was easier to learn because in programming to get the basics you need to have more lessons than the crochet lessons." These quotes suggest that drawing on crochet could provide a sampling of programming ideas to novices and helped us answer RQ3.

5.3 Non-programmers Equate Crochet to Math and Causal Relationships

Non-programmers used subfields of mathematics to describe the similarities and differences between crochet and programming. They tended to equate crochet to arithmetic or algebra, and programming to logic. When describing how they thought they performed on the crochet test, P7 stated "...I don't want to say equations, but that's the best word I can come up with..." P9 shared, "The difference is the crochet tests had more calculations. Addition, subtraction, multiplication... But the programming test was more intuitive, understanding codes, and it's logical also but there was no calculation. Both involved letters and numbers and maybe some special symbols..." P11 explained, "Crochet felt more like algebra. And the coding was kind of critical thinking or logics. And I felt like those were very different things. One's more straightforward."

All participants shared that they enjoyed learning crochet, and many of them shared that crochet reminded them of subsets of math. They characterized the two tests as being similar to different

types of math: math they either did or did not felt comfortable with. P11 further illustrated this point with the following quote: "Similarities..., like they're both kind of mathematical. So if you have a mathematics background it'd be easier to understand these things even with no coding experience or crocheting experience." These associations helped us answer RQ3.

5.4 Non-programmers Situate Their Skill Sets in Relation to Programmers

Non-programmer participants alluded to greater differences between themselves and programmers. Interestingly, when we asked these participants how they thought they performed on the programming test, five out of the six rated themselves as having done worse on the test than they actually did, providing interesting evidence to answer RQ5. We asked participants whether they believed they could get better at programming. Recalling a time they took a programming course as a degree requirement, P7 shared "I don't think my brain works that way." When asked whether they could see themselves learning programming in the future, P12 shared "On the issue of programming, I'm not absolutely sure if I would like to continue it, because I think it's more about being solitary. I don't really enjoy that, but I think I don't really have a flair for it." These quotes offered interesting insights to help us answer RQ4.

5.5 Limitations and Future Work

In the present pilot study we had a small sample size of participants. We will expand on this work by including approximately 40 more participants. It is to be expected that a pilot study of this size (12 participants) would not yield statistically significant results. We expect that the observed quantitative trends will be strengthened by statistical significance with the inclusion of more participants. Our qualitative data provided encouraging indicators of the similarities of crochet and programming and where these similarities could be applied in educational contexts.

6 Conclusion

In the present paper we explained our motivation to promote computational thinking in wider audiences. We collected both quantitative and qualitative data to provide a holistic picture of how people perceived their abilities in and related the characteristics of the art form crochet and computing. We found promising trends in the test scores of our participants, and uncovered specific mappings between elements of introductory CS and crochet. Furthermore, we discovered how those without programming experience differentiated the content of the crochet and programming tests as being similar to subsets of math.

References

[1] Sapna Cheryan, Allison Master, and Andrew N Meltzoff. 2015. Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering by diversifying stereotypes. *Frontiers in psychology* 6 (2015), 49.

[2] Mei-Hung Chiu and Jing-Wen Lin. 2005. Promoting fourth graders' conceptual change of their understanding of electric current via multiple analogies. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching* 42, 4 (2005), 429–464.

[3] Richard K Coll, Bev France, and Ian Taylor. 2005. The role of models/and analogies in science education: implications from research. *International Journal of Science Education* 27, 2 (2005), 183–198.

[4] College Board. [n. d.]. *AP Computer Science Principles*. https://apcentral.collegeboard.org/courses/ap-computer-science-principles

[5] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2022. *Introduction to algorithms*. MIT press.

[6] CrackAP.com. [n. d.]. *AP Computer Science Principles Practice Tests*. https://www.crackap.com/ap/computer-science-principles/index.html

[7] Kayla DesPortes, Kathleen McDermott, Yoav Bergner, Francisco Enrique Vicente Castro, Sauda Musharrat, and Aakruti Lunia. 2024. DanceBits 'It tells you to see us': Supporting Dance Practices with an Educational Computing Kit. In *Proceedings of the Eighteenth International Conference on Tangible, Embedded, and Embodied Interaction* (Cork, Ireland) *(TEI '24)*. Association for Computing Machinery, New York, NY, USA, Article 2, 19 pages. https://doi.org/10.1145/3623509.3633350

[8] Roy Dreistadt. 1968. An analysis of the use of analogies and metaphors in science. *The Journal of Psychology* 68, 1 (1968), 97–116.

[9] Dedre Gentner and Kenneth D Forbus. 2011. Computational models of analogy. *Wiley interdisciplinary reviews: cognitive science* 2, 3 (2011), 266–276.

[10] Arthur M Glenberg and David A Robertson. 1999. Indexical understanding of instructions. *Discourse processes* 28, 1 (1999), 1–26.

[11] Shawn M Glynn. 2012. Explaining science concepts: A teaching-with-analogies model. In *The psychology of learning science*. Routledge, 219–240.

[12] Susan Goldin-Meadow. 2009. How gesture promotes learning throughout childhood. *Child development perspectives* 3, 2 (2009), 106–111.

[13] Susan Goldin-Meadow. 2011. Learning through gesture. *Wiley Interdisciplinary Reviews: Cognitive Science* 2, 6 (2011), 595–607.

[14] Diane F Halpern, Carol Hansen, and David Riefer. 1990. Analogies as an aid to understanding and memory. *Journal of educational psychology* 82, 2 (1990), 298.

[15] Dave Heywood. 2002. The place of analogies in science education. *Cambridge Journal of Education* 32, 2 (2002), 233–247.

[16] Keith J Holyoak and Paul Thagard. 1989. Analogical mapping by constraint satisfaction. *Cognitive science* 13, 3 (1989), 295–355.

[17] Keith J Holyoak and Paul Thagard. 1996. *Mental leaps: Analogy in creative thought*. MIT press.

[18] Dan Hunter. 2004. Teaching and using analogy in law. *J. Ass'n Legal Writing Directors* 2 (2004), 151.

[19] Susan Jang, John B Black, and Robert W Jyung. 2010. Embodied cognition and virtual reality in learning to visualize anatomy. In *32nd Annual Conference of the Cognitive Science Society, Portland, OR, August*. 12–14.

[20] Hadeel Mohammed Jawad, Samir Tout, Munther Abualkibash, and Yichun Xie. 2018. Integrating art and animation in teaching computer programming for high school students experimental study. In *2018 IEEE International Conference on Electro/Information Technology (EIT)*. IEEE, 0311–0317.

[21] Donald E Knuth. 1997. *The Art of Computer Programming: Fundamental Algorithms, Volume 1*. Addison-Wesley Professional.

[22] Donald E Knuth. 2007. Computer programming as an art. In *ACM Turing award lectures*. 1974.

[23] Carly Kontra, Susan Goldin-Meadow, and Sian L Beilock. 2012. Embodied learning across the life span. *Topics in cognitive science* 4, 4 (2012), 731–739.

[24] Sheila L Macrine and Jennifer MB Fugate. 2022. *Movement matters: How embodied cognition informs teaching and learning*. MIT Press.

[25] Brian Magerko, Jason Freeman, Tom Mcklin, Mike Reilly, Elise Livingston, Scott Mccoid, and Andrea Crews-Brown. 2016. EarSketch: A STEAM-Based Approach for Underrepresented Populations in High School Computer Science Education. *ACM Trans. Comput. Educ.* 16, 4, Article 14 (Sept. 2016), 25 pages. https://doi.org/10.1145/2886418

[26] Markeya S Peteranetz, Abraham E Flanigan, Duane F Shell, and Leen-Kiat Soh. 2017. Computational creativity exercises: An avenue for promoting learning in computer science. *IEEE Transactions on Education* 60, 4 (2017), 305–313.

[27] Daniel Saunders and Paul Thagard. 2005. Creativity in computer science. In *Creativity Across Domains*. Psychology Press, 171–186.

[28] Lawrence Shapiro. 2019. *Embodied cognition*. Routledge.

[29] Hannah M Smith. 2018. *Exploring Students' Internal Motivation for Engineering Creativity: Creative Confidence and the Arts*. Master's thesis. Queen's University (Canada).

[30] Sherry Turkle and Seymour Papert. 1990. Epistemological pluralism: Styles and voices within the computer culture. *Signs: Journal of women in culture and society* 16, 1 (1990), 128–157.

[31] Margaret Wilson. 2002. Six views of embodied cognition. *Psychonomic bulletin & review* 9 (2002), 625–636.

[32] Baichang Zhong, Siyu Su, Xiaofan Liu, and Zehui Zhan. 2023. A literature review on the empirical studies of technology-based embodied learning. *Interactive learning environments* 31, 8 (2023), 5180–5199.

[33] Nikos Zourbanos, Yannis Tzioumakis, Duarte Araújo, Sofia Kalaroglou, Antonis Hatzigeorgiadis, Athanasios Papaioannou, and Yannis Theodorakis. 2015. The intricacies of verbalizations, gestures, and game outcome using sequential analysis. *Psychology of Sport and Exercise* 18 (2015), 32–41.

A Study Design Materials

A.1 Crochet Test

Gestural Pattern Knowledge Transfer for Algorithmic Languages IRB #24-942

Your participation in this study is voluntary, you are not required to participate, and you can stop participating at any time without any negative consequences to you.

Crochet Test

1. sc 3 (trc 3 hdc 3) x9

   What is the 12th stitch in the pattern above?

   a. sc
   b. trc
   c. hdc
   d. There wouldn't be a 12th stitch.

2. (S, M, L)
   dc x2 (sc 2 hdc 3) x(1, 2, 3)

   If I was making a size medium, what would my 11th stitch be?
   a. dc
   b. sc
   c. hdc
   d. There wouldn't be an 11th stitch.

3. How would I write a pattern to alternate single crochet (sc) and double crochet (dc) stitches for sizes small, medium, and large if size small took 30 stitches, size medium took 36 stitches and size large took 42 stitches?
   a.

   (S, M, L)
   (sc) dc x(30, 36, 42)

   b.

   (S, M, L)
   (sc dc) x(15, 18, 21)

   c.

   (S, M, L)
   sc (dc) x(15, 18, 21)

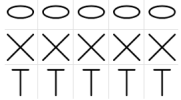
   d.

Figure 1. Three multiple choice questions on written crochet language.

Gestural Pattern Knowledge Transfer for Algorithmic Languages IRB #24-942

(S, M, L)
sc dc x(30, 36, 42)

4. How would you write the following visual pattern from top to bottom?



   a.

   ch sc hdc 25

   b.

   (ch sc hdc) x5

   c.

   ch 5
   sc 5
   hdc 5

   d.

   ch 3
   sc 3
   hdc 3

5. How can I crochet a row of 15 single crochet stitches bordered by 2 half double crochet stitches on each side?

   a.
   hdc hdc (sc) x15 hdc hdc

   b.

Figure 2. Two multiple choice questions on written crochet language.

(hdc (sc) x15 hdc) x2

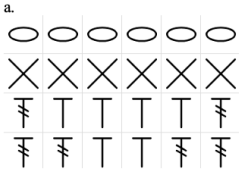
c.
(hdc hdc sc hdc hdc) x15

d.
(hdc) x2 sc (hdc) x2

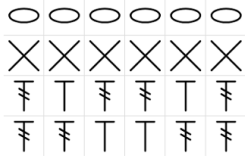
6.
What would the visual pattern be for the textual pattern written below?

ch 6
sc 6
trc (hdc) x4 trc
(trc) x2 (hdc) x2 (trc) x2

a.



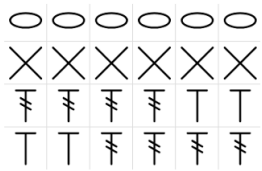
b.



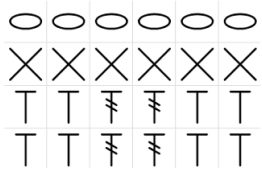
c.

Figure 3. One multiple choice question on written crochet language.

Gestural Pattern Knowledge Transfer for Algorithmic Languages IRB #24-942



d.



7.
How many stitches would this row have?

(sc 2 hdc 5) x2 (sc) x3

a. 12
b. 60
c. 23
d. 17

8.
How many stitches would this row have?

trc 2 ((sc trc) x3 (trc) x2) x2

a. 20
b. 24
c. 9
d. 18

9.
(XS, S, M, L)
trc x(4, 5, 5, 6) (sc hdc dc sc) x(12, 13, 15, 16)

If I was making a size small, what would my 11th stitch be?

Figure 4. Three multiple choice questions on written crochet language.

Gestural Pattern Knowledge Transfer for Algorithmic Languages IRB #24-942

a. trc
b. sc
c. hdc
d. dc

Figure 5. Multiple choice answers for the last written crochet language question.

A.2 Semi-Structured Interview

Questions on beliefs in abilities and success in activities:

- Did you think you could succeed on the crochet test?
  Why or why not?
  How do you think you did on the test, from 0-100%?
- Did you think you could succeed on the programming/algorithms test?
  Why or why not?
  How do you think you did on the test, from 0-100%?
- Do you think you could get better at crochet?
- Do you think you could get better at programming/algorithms?

Questions on motivation and interest in activities:

- Could you see yourself continuing to learn crochet?
  Why or why not?
- Could you see yourself continuing to learn programming/algorithms?
  Why or why not?

Questions on similarities and differences in activities:

- What differences did you see in the programming/algorithms test versus the crochet test, besides the subject matter itself?
- What similarities did you see in the programming/algorithms test and the crochet test?