

# Intersections of Literacy Acquisition and Programming

Tara Rosenberger Shankar  
MIT Media Laboratory  
20 Ames Street, E15-312  
Cambridge, MA 02139  
[tara@media.mit.edu](mailto:tara@media.mit.edu)

## 1. Introduction

The computer is a highly literate instrument. And, the design of most computer software, especially programming languages and environments, assume a high degree of literacy on the part of the user. Because most programming environments rely upon text as the primary method of specifying instructions, and because programming structures are similar in a number of ways to writing text, the development of literacy is a necessary precondition, or at least an accompanying problem when instructing low-literacy adult students in programming. An understanding of how adults acquire knowledge of and an ability to produce well-formed literate structures shares a direct relationship to their ability to program.

For the past year I have been conducting a Design Experiment [1, 2] in a basic level General Education Development (GED) course for adults in Spanish Harlem, New York City. The students have learned reading, writing, and math skills to an elementary level only. Their prospects are limited: they are too old to resume studies in the standard, age-homogenous educational environment; often too unskilled to get a minimum wage job; yet too full of hope and potential to give up on. I founded and taught a computer class that was open for anyone in the GED class to attend. I introduced commercial and innovative software into the classroom and observed the students' development in writing production.

Drawing from the past year of observing low-literacy adult students learn how to write, and, more specifically, learn how to write on computers, I discuss an area essential both to literate production and programming, the mechanics of writing. I discuss characteristics of writing produced by the adult students, and how these would impact learning to program. Lastly, I speculate on how learning to program might accelerate students' production of literate structures.

## 2. Mechanics of writing

The mechanics of writing are some of the first things a person acquires when learning to write. Forming letters by

hand; writing in a straight line from left to right; capitalizing the first letter of each sentence, acronyms and proper nouns; and putting spaces between words are some examples of written mechanics.

Like children, adult students begin to write much like they speak, transcribing their language in a slow and painstaking manner. Unlike children, most are faster using pencil and paper than a word processor because adults have mastered the rules and dexterity necessary for character formation, whereas they are less familiar with the placement of keys on the keyboard. Furthermore, not all keys make sense to them. For example, since they are coming to computers from writing (however little) on paper, they do not understand that a space, paragraph and tab are bona fide characters; they see them as undifferentiated negative space on the page. They also rarely use explicit paragraphing in their writing on computer when they may do so on paper.

Well I think that is generous thing to share with people that don't have nothing there are very poor people in different part of the world sometimes I think about those persons they probably be very hungry, cold, sick, etc. I would like to have a lot of money to help those kind of people that are having hungry everyday I don't know why the richest persons treat the others like nothing because of their money am going to give you an example what happens to me when I was working one day I was doing a video delivery for this rich house I was expecting like \$5.00 dollars tip but they only gave me \$00.50cents and I said well that's OK one day I will be like you and then I left.

### Figure 1. Student essay example.

Low-literacy adults often write without any end punctuation. Figure 1 shows a well written and lengthy example of student writing produced during class time. The entire student essay is one sentence, with a period used only to indicate the abbreviation of "et cetera". The lack of punctuation, however, is belied by the students' proper oral expression when reading their own writing. When reading out loud, they use sentence-final intonation at the points where one would expect a period. After helping them learn to punctuate their text, first by reading out loud their writing as

they have it and asking them “sounds wrong,” or having them read it and prompting them to inspect their text more closely because “they have not read what they wrote,” they will often inquire about the difference between periods and commas. From this point on, they begin to lace their writing with liberal doses of periods and commas, often incorrectly placed.

Students use capitalization unconventionally. Generally, since they do not make sentences, they do not capitalize sentence-initial letters. Figure 2 shows a shorter and less well-formed student essay produced during class time in which the student capitalizes the first letter in every word for the first paragraph, but capitalizes according to some different sense in the second paragraph. Spelling can also undergo similar variation. From word mention to mention in the same essay, the spelling may differ by one or more letters.

The mechanics absorbed through learning to write become even more critical and functional in programming. Programming requires fluency with punctuation. Commands must be terminated with semi-colons in the same way sentences are with periods. Paragraphs are formed with brackets or braces, precedence indicated with parenthesis. There must be spaces between commands. Variable names must be repeated exactly the same way each time. If a person has not learned to attend to written mechanics, programming in a textual environment will be nearly impossible.

One could consider typing an e-mail or URL address as an interpreted programming command. For example, in a typical URL, [www.yahoo.com](http://www.yahoo.com), periods are used instead of spaces between “words” – a common feature in email addresses as well. If a period is missing or incorrectly placed the browser will produce an error message, or the email will be returned to sender. A password, as well as a variable name, has a sequence of characters in which case, order and repeatability are crucial. Students have great difficulty designing, memorizing, copying down and repeating passwords. In short, without literate mechanics, adult students may not be able to make the additional leap in understanding necessary to program even URLs and email addresses successfully.

Dear Tara I Live In An Integrate Neighborhood Is  
Very Nice Area Sometimes It Get Noise

Some That Live Were am at went it to be more  
Segregat

**Figure 2. Student essay example.**

The ability to edit text for mechanical and substantive problems is required in both writing and programming. Yet, it takes months for students to learn to integrate text insertion into their writing practice. Usually, if they notice an error in

an earlier part of their writing, they will erase the entire body of text back to the error and start afresh from that point. This limits them to writing short pieces and exhausts them quickly. In Figure 2, the second paragraph’s fragmented and telescopic text is due to wave-like erasing and rewriting of text.

Navigating non-linearly through text can be seen as a symptom of refined literacy. An expert writer moves a cursor around a body of text with the intention of inserting or deleting text. Debugging requires one to change case, add punctuation, and otherwise pore over a piece of text repeatedly. At a substantive level, debugging involves reorganizing structures much like reorganizing sentences within a paragraph, or paragraphs within a text – to affect a different interpretation of the text. To emphasize the difficulty of learning to write in a non-linear way, I note that after one year of writing with word processors, not one of my students uses copy, cut and paste, though I have been teaching it since the third week of classes.

### 3. Could Learning to Program Aid Literacy?

Programming may help students grasp important literate concepts. Practice with forming small, independent phrases in an interpreted language may help students realize more quickly and fully that writing must also be terminated and punctuated to help the reader. Forced to attend to and control punctuation, spacing and case – important visual elements of literacy, students may begin to *see* these elements in their writing, even though there is no corollary in speech. The limited yet stricter programming syntax may allow students the pleasure of seeing their written expression interpreted, and the surprise of seeing it interpreted literally. This may in turn give them the experience and rationale for attending to and editing small units of text repeatedly, so necessary in writing.

### 4. References

[1] Collins, A., *Toward a Design Science of Education*, in *New Directions in Educational Technology*, E. Scanlon and T. O’Shea, Editors. 1992, Springer-Verlag: New York.

[2] Brown, A., *Design Experiments: Theoretical and Methodological Challenges in Creating Complex Interventions in Classroom Settings*. *The Journal of the Learning Sciences*, 1992. 2(2): p. (141-178).