# Exploring Challenging Variations of Parsons Problems

Nathaniel Weinman
nweinman@berkeley.edu
University of California, Berkeley

Armando Fox
fox@berkeley.edu
University of California, Berkeley

Marti Hearst
hearst@berkeley.edu
University of California, Berkeley

## ABSTRACT

Introductory programming classes teach students to program using worked examples, code tracing, and code writing exercises. Parsons Problems are an educational innovation in which students unscramble provided lines of code, as a step towards bridging the gap between reading and writing code. Though Parsons Problems have been found effective, there is some evidence that students can use syntactic heuristics to help them solve these problems without fully understanding the solution.

To address this limitation, we introduce Faded Parsons Problems, a variation of Parsons Problems where parts of the provided code are incomplete. We explore a specific instantiation of this idea, Blank-Variable Parsons Problems, in which all variable names are blanked out. Unlike another Parsons Problem variation—adding distractor code lines—Blank-Variable Parsons can be automatically created from a solution without additional effort from an instructor.

A 75 minute pilot study with CS1 students indicates that solving standard Parsons Problems does not lead to short-term near-transfer in code writing, suggesting a need for problems with less scaffolding. Additionally, students self-report Blank-Variable Parsons as fitting in difficulty between Parsons Problems and code writing, suggesting Blank-Variable Parsons may be one opportunity to fill this gap.

## CCS CONCEPTS

• **Applied computing → Interactive learning environments**;
• **Social and professional topics → CS1**.

## KEYWORDS

Parsons Problem; Faded Scaffolding; Practice Tools; CS1

## 1 INTRODUCTION AND BACKGROUND

The learning sciences suggest that students benefit from tackling problems in their *Zone of Proximal Development* (ZPD), or problems that are solvable but only with guidance or scaffolding. Parsons Problems [3], in which students unscramble provided lines of code, are an example of such scaffolding. Unlike code tracing and worked examples, they provide an opportunity for students to construct solutions, but unlike code writing, the solution space is much more limited and a lighter mastery of language syntax is required. Ericson et al. [2] found Parsons Problems to be faster than code writing exercises while producing similar learning gains in CS1 classrooms.

However, Parsons Problems have limitations as well. Denny et al. [1] raise concerns that students can "game" Parsons Problems, or make progress while avoiding learning. Additionally, though Parsons Problems support students in constructing a logical solution, students do not write any code. We introduce and study **Blank-Variable Parsons Problems**, a more challenging variant in which all variable names are removed from the provided lines.

## 2 METHODS AND RESULTS

We ran a pilot study in the summer of 2019 with CS1 students at a large US university. 13 participants (11 Male) were compensated monetarily for their time and offered a chance to review Multiple Recursion. The researchers were not course instructors.

Participants worked through a total of 4 exercises. The first 3 exercises consisted of two challenging Multiple Recursion questions and one easy question (Fizz Buzz) in a fixed order. Participants were randomly assigned, without replacement, to work on these three exercises in each of the three problem type: Parsons Problems, Blank-Variable Parsons, and code writing. The final exercise repeated the first question, but was always code writing.

Though all participants working on the first Multiple Recursion exercise as a standard Parsons Problem were able to solve it, none were able to solve the same problem at the end of the study as a code writing exercise. Additionally, at the end of the study, participants self-rated Blank-Variable Parsons as harder to solve than Parsons Problems but easier to solve than code writing questions.

## 3 CONTRIBUTIONS AND FUTURE WORK

This pilot study provides early motivation for the opportunity of further fading between Parsons Problems and code writing. It introduces Blank-Variable Parsons, a more challenging variation of Parsons Problems, as a problem which may serve this need. It more broadly opens up a discussion for other instantiations of Faded Parsons Problems, in which parts of the provided code are incomplete.

## ACKNOWLEDGMENTS

This material is based upon work supported by by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 1752814 and the Google Social Interactions Research Award.

## REFERENCES

[1] Paul Denny, Andrew Luxton-Reilly, and Beth Simon. 2008. Evaluating a new exam question. In *ICER '08*.
[2] Barbara J. Ericson, Lauren E. Margulieux, and Jochen Rick. 2017. Solving parsons problems versus fixing and writing code. In *Koli Calling '17*.
[3] Dale Parsons and Patricia Haden. 2006. Parson's Programming Puzzles: A Fun and Effective Learning Tool for First Programming Courses *(ACE '06)*. 157–163.

ACM ISBN 978-1-4503-6793-6/20/03.
https://doi.org/10.1145/3328778.3372639