Conducting Multi-Institutional Studies of Parsons Problems

Barbara J. Ericson[*]
barbarer@umich.edu
University of Michigan
Ann Arbor, MI, USA

Janice L. Pearce[*]
jan_pearce@berea.edu/jpearce@ashesi.edu.gh
Berea College / Ashesi University
Berea, KY USA / Accra, Ghana

Susan H. Rodger[*]
rodger@cs.duke.edu
Duke University
Durham, NC, USA

Andrew Csizmadia
a.p.csizmadia@newman.ac.uk
Newman University
Birmingham, England, UK

Rita Garcia
rita.garcia@vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

Francisco J. Gutierrez
frgutier@dcc.uchile.cl
DCC, University of Chile
Santiago, Chile

Konstantinos Liaskos
k.liaskos@strath.ac.uk
University of Strathclyde
Glasgow, Scotland, UK

Aadarsh Padiyath
aadarsh@umich.edu
University of Michigan
Ann Arbor, MI, USA

Michael James Scott
michael.scott@falmouth.ac.uk
Falmouth University
Cornwall, UK

David H. Smith IV
dhsmith2@illinois.edu
University of Illinois
Urbana-Champaign, IL, USA

Jayakrishnan Warriem
jkm@nptel.iitm.ac.in
Indian Institute of Technology Madras
Madras, India

Angela Zavaleta Bernuy
angelazb@cs.toronto.edu
University of Toronto
Toronto, Canada

ABSTRACT

Many novice programmers struggle to write code from scratch and get frustrated when their code does not work. Parsons problems can reduce the difficulty of a coding problem by providing mixed-up blocks that the learner assembles in the correct order. Parsons problems can also include distractor blocks that are not needed in a correct solution, but which may help students learn to recognize and fix errors. Evidence indicates that students find Parsons problems engaging, easier than writing code from scratch, useful for learning patterns, and typically faster to solve than writing code from scratch with equivalent learning gains. This working group leverages the work of the 2022 ITiCSE working group which published an extensive literature review of Parsons problems and designed and piloted several studies based on the gaps identified by the literature review. The 2023 working group is revising, conducting, and creating new studies. We will analyze the data from these multi-institutional and multi-national studies and publish the results as well as recommendations for future working groups.

CCS CONCEPTS

• **Social and professional topics** → *Computing education.*

KEYWORDS

Parsons Problems, Parsons Puzzles, Parson's Programming Puzzles

ACM Reference Format:
Barbara J. Ericson, Janice L. Pearce, Susan H. Rodger, Andrew Csizmadia, Rita Garcia, Francisco J. Gutierrez, Konstantinos Liaskos, Aadarsh Padiyath, Michael James Scott, David H. Smith IV, Jayakrishnan Warriem, and Angela Zavaleta Bernuy. 2023. Conducting Multi-Institutional Studies of Parsons Problems. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 2 (ITiCSE 2023), July 8–12, 2023, Turku, Finland.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3587103.3594211

1 INTRODUCTION

Programming courses often require students to write code from scratch, but the difficulty of this task can overwhelm novice students [1, 5]. One recommended way to reduce the difficulty of whole tasks such as writing code from scratch is to use completion problems such as Parsons problems [8]. In Parsons problems learners arrange mixed-up fragments in the correct order to solve a problem [2]. The problems may also include extra blocks, called distractors, that are not needed in the solution. Parsons problems were named for Dale Parsons who published one of the early studies on them in computing education with Patricia Haden [6]. They thought that Parsons problems would be engaging since they were puzzle-like, that distractors would help students learn to avoid or fix common syntax and semantic errors, and that the solved problems would serve as examples of well-written code. They provided evidence that most students found solving Parsons problems useful for learning to program [6]. The growing research on Parsons problems has also provided evidence for several benefits from Parsons problems [2] including engagement [4], learning efficiency [3, 10], and recalling programming patterns [9]. However, there is a clear need for replication studies and more multi-institutional studies since most of the research has been single institution studies of less than 500 participants.

[*] co-leader

ITiCSE 2023, July 8–12, 2023, Turku, Finland

2 OBJECTIVES

The 2022 ITiCSE working group on Parson's problems conducted an extensive literature review and designed and piloted several studies based on issues raised by the literature review [2]. We have revised these studies and the research questions and are in the process of conducting studies with the goal of better understanding the effectiveness of Parsons problems in multiple contexts. Several of the working group members have already received approval to conduct the studies with human subjects and are currently running one or more studies. Some members have access to students in the months before the conference and will run the studies then. Others will contribute by conducting think-aloud observations to gather more in-depth feedback and/or by helping analyze the data.

3 STUDIES

All studies begin with an introduction page that describes the parts of the study and the estimated completion time. The studies include a pre-survey with self-efficacy questions as well as questions where students rate their familiarity with the concepts covered by the study. Since many of the institutions do not use the study platform, the open-source ebook platform Runestone, there are video introductions to the types of problems in the study and practice problems to check that students can effectively solve these types of problems. All studies include a post-test. There is also an optional pre-test and optional post-survey to gather more information on the study participants.

There are three between-subject studies in which participants are randomly assigned to one of two conditions. These studies compare the learning performance on the post-test between the conditions. The three studies *p3pt*, *class-exp*, and *class-tog* take from 50 to 75 minutes to complete. In *p3pt* the conditions are solving Parsons problems with distractors versus writing code from scratch. In *class-exp* the conditions are solving Parsons problems with distractors versus no distractors. In *class-tog* the conditions are solving write code problems with Parsons problems as scaffolding versus write code problems with no scaffolding.

The 2022 working group also created a 30 minute study *python-swap*, to test the hypothesis that solving Parsons problems can help students learn to write code for common algorithms. In *python-swap* students solve a series of Parsons problems with the first containing blocks with comments that describe the algorithm for swapping the values of two variables, the second containing blocks with both the comments and code, and the third with blocks that only contain the code. The post-test asks students to write the swap code for two isomorphic problems. We plan to create a similar study using the rainfall problem, which has been used as an assessment in computing education [7].

We also identified the need for another study with and without distractors for students with more experience in Python and are currently developing that study.

4 CONTEXT OF STUDIES

Our studies of Parson's problems will be conducted in at least thirteen institutions situated in at least eight countries. In particular, we are conducting studies at the University of Toronto in Canada, the University of Chile in Chile, Falmouth University and Newman University in England, Ashesi University in Ghana, Indian Institute of Technology Madras in India, the University of Canterbury and Victoria University in New Zealand, University of Strathclyde in Scotland, and Berea College, Duke University, the University of Illinois, and the University of Michigan in the USA.

The study settings are diverse beyond institution and country. They are being conducted at a variety of institution types from small liberal arts teaching-focused schools to large research universities. The course modality ranges from in-person to online. Studies are being conducted in the language of instruction which includes both English and Spanish. The proportion of international students in the course ranges from none to around a third. The gender breakdown in the courses range from 50% female and 50% male to a small percentage of females. The type of courses include introductory programming (CS1) courses which primarily focus on teaching programming to computing majors to courses that serve multiple majors in which programming is only a component of the course. Student backgrounds range from novice programmers with no prior programming experience to intermediate programmers who have a semester or more of prior experience. Course sizes range from less than 30 students to more than 200, and the number of sections of a course where the study is being administered ranges from one to six. Prior familiarity with Parsons problems varies from no prior experience to significant exposure in the same e-book platform as the studies.

REFERENCES

[1] Klara Benda, Amy Bruckman, and Mark Guzdial. 2012. When life and learning do not fit: Challenges of workload and communication in introductory computer science online. *ACM Transactions on Computing Education (TOCE)* 12, 4 (2012), 1–38.

[2] Barbara J Ericson, Paul Denny, James Prather, Rodrigo Duran, Arto Hellas, Juho Leinonen, Craig S Miller, Briana B Morrison, Janice L Pearce, and Susan H Rodger. 2022. Parsons Problems and Beyond: Systematic Literature Review and Empirical Study Designs. *Proceedings of the 2022 Working Group Reports on Innovation and Technology in Computer Science Education* (2022), 191–234.

[3] Barbara J Ericson, Lauren E Margulieux, and Jochen Rick. 2017. Solving parsons problems versus fixing and writing code. In *Proceedings of the 17th Koli Calling International Conference on Computing Education Research.* 20–29.

[4] Kyle J Harms, Noah Rowlett, and Caitlin Kelleher. 2015. Enabling independent learning of programming concepts through programming completion puzzles. In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC).* IEEE, 271–279.

[5] Paivi Kinnunen and Beth Simon. 2010. Experiencing programming assignments in CS1: the emotional toll. In *Proceedings of the Sixth international workshop on Computing education research.* 77–86.

[6] Dale Parsons and Patricia Haden. 2006. Parson's programming puzzles: a fun and effective learning tool for first programming courses. In *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52.* 157–163.

[7] Otto Seppälä, Petri Ihantola, Essi Isohanni, Juha Sorva, and Arto Vihavainen. 2015. Do we know how difficult the rainfall problem is?. In *Proceedings of the 15th Koli Calling Conference on Computing Education Research.* 87–96.

[8] Jeroen JG Van Merrienboer and John Sweller. 2005. Cognitive load theory and complex learning: Recent developments and future directions. *Educational psychology review* 17, 2 (2005), 147–177.

[9] Nathaniel Weinman, Armando Fox, and Marti A. Hearst. 2021. Improving Instruction of Programming Patterns with Faded Parsons Problems. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21).* Association for Computing Machinery, New York, NY, USA, Article 53, 4 pages. https://doi.org/10.1145/3411764.3445228

[10] Rui Zhi, Min Chi, Tiffany Barnes, and Thomas W Price. 2019. Evaluating the Effectiveness of Parsons Problems for Block-based Programming. In *Proceedings of the 2019 ACM Conference on International Computing Education Research.* 51–59.